R&D 5366-EN-01
DAJA45-89-C-0004. 2

On-line point positioning with single frame camera data

Final report

Th. Kersten, K. R. Holm, A. Gruen
Institute of Geodesy and Photogrammetry
ETH-Hönggerberg, CH - 8093 Zürich
Switzerland

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188
Exp Date Jun 30, 1986

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) R&D 5366-EN-01
6a. NAME OF PERFORMING ORGANIZATION Institute of Geodesy & Photogrammetry 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION USARDSG-UK
6c. ADDRESS (City, State, and ZIP Code) ETH-Hoenggerberg, CH-8093 Zurich Switzerland	7b. ADDRESS (City, State, and ZIP Code) PSC 802 BOX 15 FPO AE 09499-1500
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Army Research Office 8b. OFFICE SYMBOL (If applicable) ARO	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DAJA45- 89-C-0004
8c. ADDRESS (City, State, and ZIP Code) PO Box 12211 Research Triangle Park, NC 27709-2211	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO.: 61102A PROJECT NO.: 1L161102BH57 TASK NO.: 01 WORK UNIT ACCESSION NO: AR

11. TITLE (Include Security Classification)
(U) ON-LINE POINT POSITIONING WITH SINGLE FRAME CAMERA DATA

12. PERSONAL AUTHOR(S)
PROFESSOR A. GRUEN

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
FINAL	FROM 01.01.89 TO 31.12.91	1992 - 03 - 15	24

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	On-line triangulation, initial value computation, sequential and simultaneous adjustment, blunder detection
0805			

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Within Final Report

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☒ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL JERRY C. COMATI	22b. TELEPHONE (Include Area Code): 071-402-7331/8490 22c. OFFICE SYMBOL: AMXSN-UK-RE

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

On-line point positioning with single frame camera data

Final technical report

by

Th. Kersten, K. R. Holm, A. Gruen

March, 1992

United States Army
EUROPEAN RESEARCH OFFICE OF THE U.S. ARMY

London England

CONTRACT NUMBER .... DAJA 45-89-C-004

Institute of Geodesy and Photogrammetry
ETH-Hönggerberg, CH - 8093 Zürich
Switzerland

Table of Contents

Table of Contents

On-line point positioning with single frame camera data

Final report

Th. Kersten, K. R. Holm, A. Gruen
Institute of Geodesy and Photogrammetry
ETH-Hönggerberg, CH - 8093 Zurich
Switzerland

Summary

The project "On-line point positioning with single frame camera data" was concluded on December 31, in 1991 after a period of three years. This final report presents the investigations and results of this project. The goal of the project was to develop a data processing concept for on-line high accuracy point positioning system. An operational on-line system, installed in a UNIX environment under the programming language "C", for aerial phototriangulation working on an analytical plotter or on a digital workstation should provide high speed of operation, accuracy of results, and ease of data management. Established tests showed that the Sun Microsystems workstations have sufficient memory capacity and speed performance to run on-line triangulation applications. The adopted algorithms in the on-line triangulation system for the computation of initial values, for the sequential and simultanous adjustment and for quality control will be presented in this report. A description of the developed software modules and the program will also be given. Although this system was designed and developed specifically for aerial triangulation, it is anticipated that future applications for on-line triangulation algorithms and methods will be found in robotics and industrial quality control.

1. Project data

The project has been defined as "On-line point positioning with single frame camera data". The research reported in this final report has been made possible through the support and sponsorship of the U.S. Government, Department of Defense, through its European Research Office of the U.S. Army in London. The period for research and investigations was fixed at 3 years, from January 1, 1989 to December 31, 1991. Researchers on the project have been Dr. Knut R. Holm as Postdoctoral Researcher from April 1, 1989 to March 31, 1990 and Thomas Kersten as Research Associate from May 1, 1989 to December 31, 1991. The investigations, implementations and results of this project have been described six Interim Reports, which are listed in section 10.1. (see Project References). The final report summarizes all work undertaken in this project.

2. Introduction

Aerotriangulation is an efficient photogrammetric method of point determination with large data sets and plays an important role in surveying. To improve the potential efficiency, accuracy and reliability of off-line aerotriangulation, the operator should receive a continuous information feedback on the progress and quality of his measurements/work. In this context, measurements in large photogrammetric blocks require quality control at an early stage of data acquisition. While, in general, the procedures of the conventional off-line aerotriangulation are separated in data preparation, data acquisition, bundle adjustment, blunder detection and location, and finally quality control, the on-line solution integrates all these data processing phases into one process. Because of the advantages of the sequential solution, on-line data processing has been an important issue in photogrammetry for many years. In general terms, "on-line triangulation" is the procedure of measuring and immediately processing data for point positioning purposes. "On-line" means direct communication with a computer during the triangulation procedure, so that measurements be immediately processed after being acquired.

After the foundation of a working group "Analytical On-line Triangulation" (WG III/4) at the XIIIth ISP Congress in Helsinki 1976 the term On-line Triangulation (OLT) has been defined in the resolution T III/3 of the XIVth Congress of ISPRS in Hamburg 1980, because:

> *" ... the on-line capability to measure and immediately process data increases the speed and reliability of photogrammetric triangulation and may significantly improve the organization of routinely performed work."*

An increase of speed and reliability of the overall triangulation procedure can primarily be achieved by controlling blunders and removing false, or adding new observations at an early stage. In order to perform these functions, an OLT system requires effective hardware and software for processing large data sets in near real time. Essentially then, the performance of such systems depends on the efficient tuning of algorithms, hardware and software. For this reason, full utilization of the potential provided by the hardware always depends on efficient software. The most time consuming aspect in OLT is the adjustment of the data. Consequently, fast sequential algorithms have been suggested for adjusting photogrammetric measurements. Furthermore, the computation of initial values as discussed in section 5.1 and the updating of the covariance matrix of the residuals (Q_{vv}) must be solved efficiently (section 5.3).

Simply stated, the realizing of an OLT system requires powerful computer hardware and optimal algorithms.

3. Goals of the project

The purpose of this project was to develop a data processing concept and a software implementation for an on-line high accuracy point positioning system. To this end an operational on-line system for aerial phototriangulation, whose performance would considerably exceed the currently available systems with respect to speed of operation, accuracy of results, and ease of data management was to be built. The system should be capable of monitoring on an analytical plotter or on a digital workstation using digital images. It was recognised that investigations into

- Initial value computations,
- Sequential and simultaneous bundle adjustment,
- Blunder detection and
- Self calibration

would be needed. Furthermore, the option to expand the on-line system by the use of kinematic GPS-data and automatic point transfer was to be considered. Later the system may be modified to be effectively used in terrestrial applications (close-range photogrammetry) as well. This expansion to a more general on-line triangulation system requires, in particular, further investigations into initial value computations. These goals are further eloborated upon the project in the initial proposal. It should be noted that it was necessary to change some of these goals or their priorities and to introduce new goals due to experiences made in developing the on-line triangulation system **OLTRIS**. For example, after discussions with the project sponsor, a higher priority was given to building an operational user interface based on a Window System before implementing self calibration.

4. Hardware aspects

Before developing and implementing an OLT system, computing hardware needs to be considered in order that acceptable performance for the OLT applications can be achieved. The computing environment at the Institute of Geodesy and Photogrammetry consists of a network of Sun Microsystems workstations. Consequently, OLTRIS has been developed and implemented in a UNIX environment (SunOS Release 4.1.1) on a Sun workstation in the programming language "C". It was assumed that memory capacity of the system be practically unlimited with respect to the number of image points, object points, photographs and cameras in the software. Moreover, response times should be well within the limits of the operator's patience. Therefore, at the commencement of this project the necessary efficiency and capacity of the SUN workstation was evaluated. The possible memory requirements are given in the following example. For a normal equation system based on observation data of 50 photos and 500 object points, each measured in 4 images, 2 Mb are required, while 100 photos and 1000 object points need 5 Mb. This data resides in memory during sequential adjustment. Also, for digital images (or windows thereof) available for on-screen viewing, 10 image patches of 512 x 512 at 8-bit resolution pixels occupy 2.5 Mb. However, only a few need to be in memory at any given time. Furthermore, for digitized points, which can be defined by small patches (less than 25 x 25 pixels), 1000 image patches require about 0.5 Mb. Again, not all would be required to be kept in memory at any time. Using the workstations of SUN Microsystems these requirements of memory capacity and speed are/canbe met. The efficiency of three generations of standard SUN workstations is given in table 1. In this context, the speed of the computer is judged by "Million Floating

Point Operations Per Second" (MFLOPS) and by "Million Instructions Per Second" (MIPS), using double-precision LinPack as reference.

workstation	speed		frequency	capacity	
	[MFLOPS]	[MIPS]	[MHz]	[Mb] minimal	[Mb] maximal
SUN 3/110	0.4	2.0	16.7	12	12
SUN SPARCstation 1+	1.7	15.8	25.0	16	16
SUN SPARCstation 2	4.2	28.5	40.0	16	96

Table 1: Performance of SUN workstations

Specific examples of the performance of a SUN workstation for OLT applications are presented below in section 5.2.2 and 5.3.1.

5. Adopted algorithms

Of particular research interest are the algorithms used for computation and providing initial values for the unknown parameters, for the procedures of the adjustment and for the quality control of the measurements. The algorithms used in OLTRIS are described below.

5.1. Computation of initial values

For simultaneous and sequential adjustment OLTRIS has to provide initial values for the unknown parameters, e.g. object point coordinates, exterior orientation parameters. The adopted approaches for computing initial values are summarized in the following.

5.1.1. Space resection

The spatial position of the camera and the rotation angles of its optical axis during the moment of exposure can be determinated by space resection. Thus, space resection is a method of computing the six exterior orientation elements (X_0, Y_0, Z_0, φ, ω, κ) of one photograph. Mathematical formualtion of space resection is based on *the collinearity equations*. The geometric principle of these equations is that an object point P_i, its image point P'_i and the perspective center O are collinear (Figure 1).

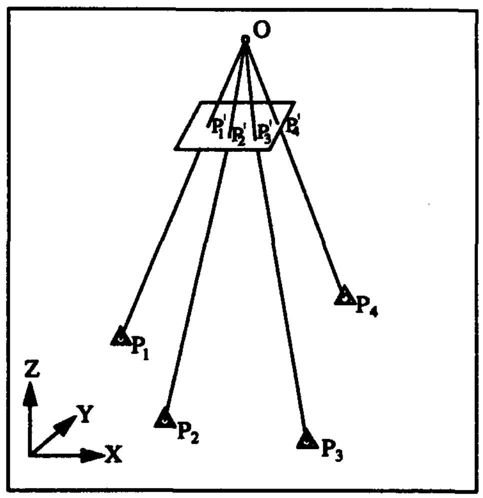

Figure 1: Geometry of space resection

The collinearity equations are expressed by the following formulas

$$F_x = (x - x_p) = -c \frac{U}{W}$$
$$F_y = (y - y_p) = -c \frac{V}{W} \qquad (1)$$

where

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = D \begin{bmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{bmatrix}$$

and:

x, y .... image coordinates,
x_p, y_p, c .... elements of interior orientation,
X, Y, Z........... object coordinates of point P_i,
D .... 3 by 3 orthogonal rotation matrix and
X_0, Y_0, Z_0 ..... object coordinates of perspective center O.

The elements of interior orientation (x_p, y_p, c) define the location of the center of perspective in the image coordinate system (physically the front nodal point) and they are given in the camera calibration data file of OLTRIS. The exterior orientation elements define the location of the perspective center in object space (X_0, Y_0, Z_0) and the orientation of the image coordinate system with respect to the object coordinate system (three rotations φ, ω, κ). For the computation of the exterior orientation parameters at least three known object points must appear in the image. The image point coordinate observations of the object points yield six collinearity equations to compute the exterior orientation elements without redundancy. To exploit any redundancy in the observations, least squares adjustment is commonly employed.

Space resection will not yield a solution when all image points lie on a straight line or too tightly grouped. To avoid these cases, the system tests the configuration of the available points in image space before the adjustment. The test criterion used is the area formed by the object points in image space. The area formed by three points will be computed as a triangle (see Figure 2). If there are more than three points available, the area will be computed

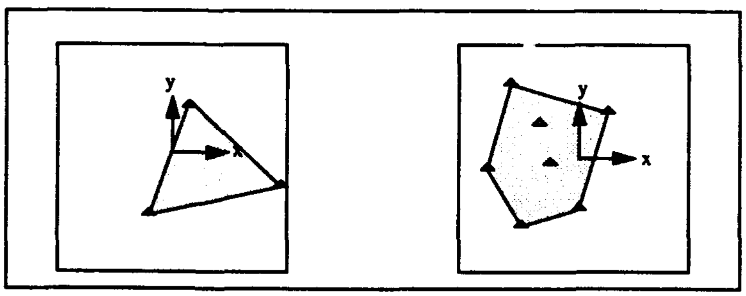

Figure 2: Area of a triangle in image space (left) and of a group of points in image space (right)

as the boundary of the extreme points (see Figure 2). The test criterion is 5% of the area of an image, e.g. for an image 23 cm x 23 cm it is 26.45 cm^2. If the computed area is under this threshold, the resection will not be computed and the system will suggest to the operator to measure more known points or to perform a relative orientation of the images .

For the least squares adjustment of the space resection starting values (approximations) of the unknown parameters are necessary. As a start, X_0 and Y_0 are the X- and Y-components of the gravity center of the object coordinate system. The height of the projection center Z_0 can be approximated with the formula

$$Z_o = Z_{os} + c\frac{d_o}{d_i} \qquad (2)$$

with:

Z_0 .... height of the projection center,
Z_{0S} .... height of gravity center in object coordinate system,
c..................... camera constant,
d_0................... averaged distance in object space and
d_i .... averaged distance in image space.

Initially, the rotation angles are set to $\varphi = \omega = 0$, with κ computed as

$$\kappa = \arctan\frac{x_{is}}{y_{is}} - \arctan\frac{X_{os}}{Y_{os}} \qquad (3)$$

with:

κ.................... rotation angle,
x_{is}, y_{is}............ coordinates of gravity center in image space and
X_{0S},Y_{0S}......... coordinates of gravity center in object coordinate system.

Using these approximations, the exterior orientation elements will be computed in an least squares adjustment of the resection.

5.1.2. Relative orientation

If the exterior orientation elements of two photographs can not be determinated by resection in space, the relative position and attitude of the images in a stereoscopic pair with respect to each other can be computed. The two photographs are orientated relatively to each other if all corresponding pair of rays from the two images to the object points intersect in space. Because of model deformations, the exact intersection in space cannot be made in all corresponding pairs of rays. For relative orientation, a minimum of five corresponding pairs of rays are needed, while a sixth pair is useful for checking. In OLTRIS two methods, the *swing-swing* method and the *one-projector* method, are used for relative orientation. Both methods are based on the *coplanarity equation*. Geometrically the coplanarity equation represents a plane which is formed by two conjugate image points, the perspective centers and the object point all form (condition b in Figure 3). To realize the coplanarity condition, the volume of the tetrahedra, as depicted in condition a of figure 4, must be zero. The formulas of the coplanarity equation are described in Karara (ed.)[1].

[1] Non-Topographical Photogrammetry, American Society of Photogrammetry, 1989, page 45/46.

The swing-swing method determinates two rotation angles (φ, κ) of the left image and three rotation angles (φ, ω, κ) of the right image, with the other orientation parameters being fixed. This method will be only used in OLTRIS for computing the rotation angles of the first model. The one-projector method determinates the y- and z-base components, and the rotation angles of the second image, with the first seven orientation parameters, six of the first image and one of the second image, being fixed. If redundant image point observations are measured, the orientation is computed by a least squares adjustment.

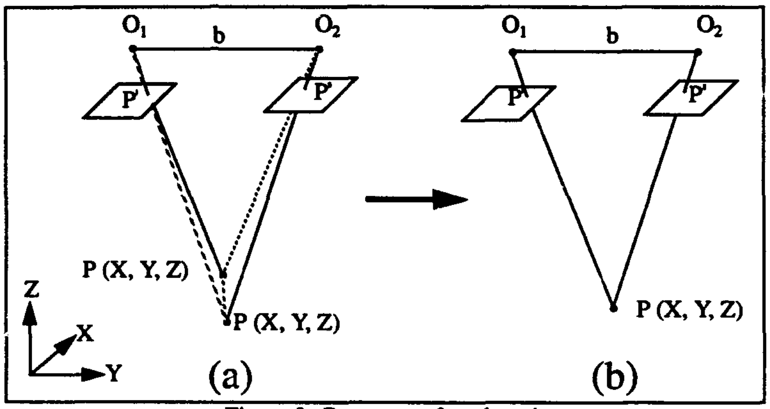

Figure 3: Geometry of coplanarity

5.1.3. Spatial intersection

If the exterior orientation elements of two photographs in a overlapping pair (stereopair) are known, the spatial position (X, Y, Z) of any point can be determinated from the measured photo coordinates in the two images using the method of spatial intersection. The geometric principle (Figure 4) is based on a spatial intersection of two rays from two fixed points (camera stations O_1, O_2).

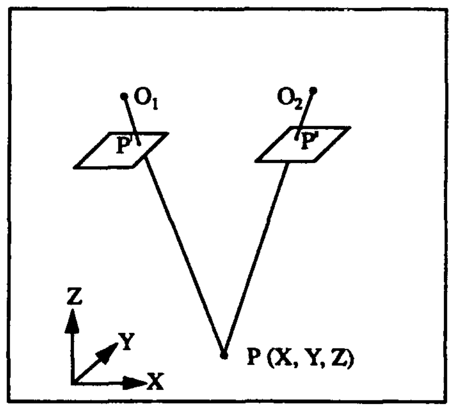

Figure 4: Geometry of spatial intersection

The formulas for spatial intersection used in OLTRIS, are essentially the same as shown for this purpose in Karara (ed)[2].

[2] Handbook of Non-Topographical Photogrammetry, American Society of Photogrammetry, 1979, page 85 - 89

The method gives initially four equations for solving three parameters (X, Y, Z). Those four equations can be combined in three different ways, each giving three linear equations. In Karara, one such combination is chosen, giving one set of equations. Then the three equations are combined, giving a "closed solution" for X, Y, and Z, i.e. one formula for each parameter (X = f(observations),...). The formulation fails in the event of an "unfortunate" base direction, due to a denominator approaching zero. The algorithm in OLTRIS therfore contains all the three sets of closed solution formulas. The used set of the appropriate formulas, chosen to give best precision in the result, has the largest denominator.

The computed object point coordinates are located in the coordinate system fixed by the exterior orientation parameters of the images.

5.1.4. Three-dimensional similarity transformation

Because of insufficient datum information during the triangulation processing, the object points and the perspective centers must be defined in a local coordinate system. Later, a three dimensional coordinate transformation can be used to transfer the local into a global datum as long as a minimum of three identical points in both coordinate systems are present. Within OLTRIS, a three-dimensional similarity transformation transforms a local into a global coordinate system, which is defined by the ground control points. If insufficient control point information is available, a local coordinate system is fixed at the start of the triangulation. The formulas used for the transformation are the same as described in Albertz/Kreiling[3]. The parameters of the transformation to be determinated are a uniform scale factor, three rotations and three translations. These unknowns can be obtained by a standard least squares adjustment. Note that a minimum of three points should be known in each coordinate system - two X-, Y-coordinates and three Z-coordinates. The three-dimensional similarity transformation provides OLTRIS with the flexibility to start the triangulation in a local coordinate system and to transform it to the global system whenever sufficient control point information is available.

5.1.5. Strategy of providing initial values for on-line data processing

A special problem at the start of the photogrammetric triangulation is to provide initial values for the unknown parameters (i.e. object point coordinates and exterior orientation elements) of least squares adjustment. Herein, the choice of functions for computing initial values depends on the given information. Therefore, the triangulation procedure is based on one of the following cases:

- exterior orientation elements given, e.g. determined by General Positioning System (GPS), no control points (rare case).
- some control points given, determined by geodetic network adjustment; no known exterior orientation elements given (normal case).
- control points and known exterior orientation elements given (ideal case).
- no information from a ground coordinate system; global transformation impossible (bad case).

[3] Photogrammetric Guide, Wichmann Verlag, Karlsruhe 1980, page 45 - 47.

In each case, a strategy for providing the necessary initial values is needed. For this reason, besides the coding of functions for the computation of initial values, the investigations of the project have also concentrated on finding the best strategy of providing initial values for on-line data processing. A good strategy is characterized by the saving of process time and by a minimum of user interaction, such as for for choosing the method to be used or, judging the results of the computations. So, for the rare case of having known exterior orientation elements of all camera stations only spatial intersections for determining object point coordinates are used. But if only some control points and no known exterior orientation elements are available (normal case) in the photogrammetric block, the triangulation has to start in a local coordinate system by orienting the first two images relatively. The exterior orientation elements of the following consecutive images in the strip can be determined by space resection if sufficient tie points can be measured. The transformation into the global net can be achieved when sufficient control point information is available (i.e. for the three-dimensional similarity transformation). In an ideal case, as mentioned, when sufficient control points and known exterior orientation elements are given, only spatial intersections have to use for computing object point coordinates of additional natural points. If no information of a global net is available (bad case), the triangulation has to be processed in a local datum. Herein, the same strategy, as used for a global datum, for providing initial values will be used except that a transformation is not performed.

OLTRIS computes initial values automatically after each measurement, and before updating the normal equation system with any new observations, if sufficient information for computing initial values is present. On the other hand, the necessary initial values of all new observations will be computed, if the operator selects the option *Display* or *Update* on the on-line triangulation processing menu (see Figure A in Appendix A). If the computation of initial values is not possible due to insufficient information, the observations to be included receive the status *waiting* or *not included.*

The fast and precise computation of initial values improves the performance of an on-line triangulation and data processing system like OLTRIS.

5.2. Adjustment

Since near real-time responses are crucial factors in on-line triangulation, fast sequential adjustment algorithm is essential. As each estimation algorithm can be reformulated into a sequential mode, a great variety of sequential techniques to be considered arises. The important sequential algorithms and also a brief comparison thereofis presented in section 5.2.1. The conclusion was to implement the Givens Transformations (GT) algorithm for sequential adjustment. Because of avoiding the time consuming inversion of a normal equation system, the Cholesky factorization and back substitution for the relinearizing and update of the parameter vector was implemented, for the simultaneous least squares adjustment case.

5.2.1. Simultaneous adjustment by Cholesky factorization and back substitution

In order to adjust the observations simultaneously the Cholesky algorithm was implemented in OLTRIS. This algorithm relinearizes the normal equation system and updates the parameter vector.

In the bundle adjustment it is assumed that the linearized observation equation system is

$$v = Ax - l; P \tag{4}$$

with:

v .... vector of residuals,
A .... design matrix,
x .... vector of unknown parameters,
l .... vector of observations and
P .... weight matrix.

This equation system is solved by the least squares method, by building and solving the normal equation system (5),

$$A^T P A x = A^T P l \tag{5}$$

or

$$N x = n \tag{6}$$

with:

N .... normal equation matrix and
n .... vector of the right-hand-side.

The explicit solution to the unknown parameters may be written as

$$x = N^{-1} n = (A^T P A)^{-1} A^T P l. \tag{7}$$

Instead of direct inversion, the Cholesky's algorithm decomposes the normal equation matrix N into a lower triangular matrix C and an upper triangular matrix C^T

$$N = C C^T. \tag{8}$$

Further, the vector of the right-hand-side, n can be decomposed as

$$n = C c \tag{9}$$

The reduction (factorization) of the normal equation matrix N yields the upper triangular matrix C^T and vector c. The unknown vector x is solved by back substitution using

$$C^T x = c. \tag{10}$$

Figure 5 displays the main steps of Cholesky factorization and back substitution graphically.

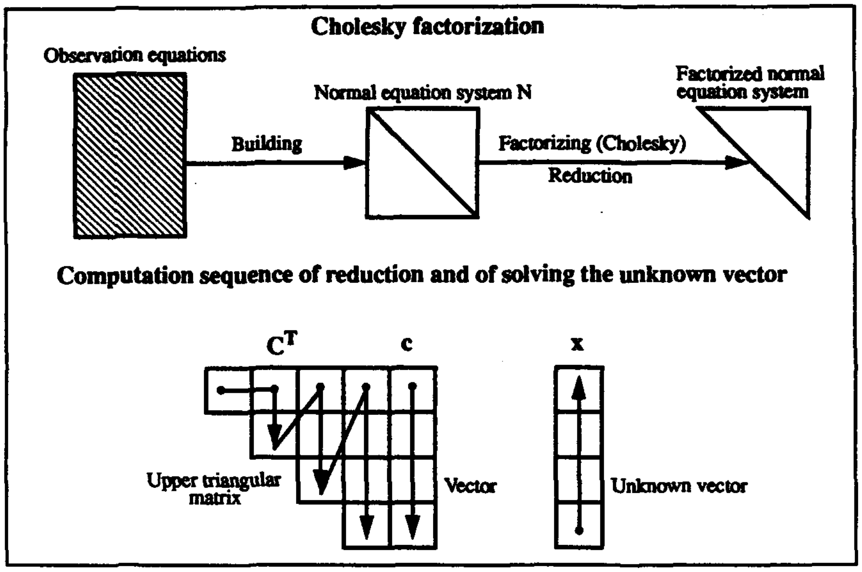

Figure 5: Relinearization of the normal equation system and update of parameter vector by Cholesky algorithm

5.2.2. Sequential adjustment by Givens Transformations

A number of algorithms for sequential adjustment have been suggested. Mikhail and Helmering (1973) present methods are based on updating the inverted normals directly. This method is also known as "stationary Kalman filtering" and has been adopted for OLT by several authors [Helmering, 1977; Kratky, 1982; Dowideit, 1980, 1982]. Gruen (1982) recommended the Triangular Factor Update (TFU), which updates the factorized normals based on Gauss or Cholesky decomposition. The procedure of TFU is also described by Wyatt (1982). He compared the Kalman-form to the TFU algorithm. The conclusion of the comparison was that the TFU algorithm is superior to the Kalman-form algorithm both with respect to time consumption and storage requirements. The use of the Givens Transformations in photogrammetric applications has been suggested by Blais (1983).

The relative performance of these sequential algorithms can best be judged by practical tests using software which is as close as possible to a real application. Such tests were carried out in on-line phototriangulation by Runge (1987) and Holm (1989c). They describe the use of GT in on-line phototriangulation and compare this approach to the TFU algorithm. Runge's results of the comparison with the TFU method showed that only 40% of the time is needed for the update of the reduced normal equations when using GT. In Holm's test, the GT algorithm appears to be up to four times faster than the TFU for updating the factorized normal equation system.

The performance of GT in sequential estimation will be given in the following. For these investigations the testfield block Echallens, with a total of 49 photograph and 231 object points, was used. The computation was made on a SUN 3 computer using a 68020 CPU

with floating point accelerator. (Note that the computation on a SUN SPARCstation 1+ is 30% faster than on the SUN 3). Figure 7 illustrates the results of the computations of the solution vector in a sequential mode when building up the block from 2 to 49 photographs. Herein, groups of measurements have been introduced at different stages ranging from 34 to 50 image points per update. The abscissa gives the stage of the sequential process (number of photographs) while the ordinate gives the computing time per new image point in seconds.

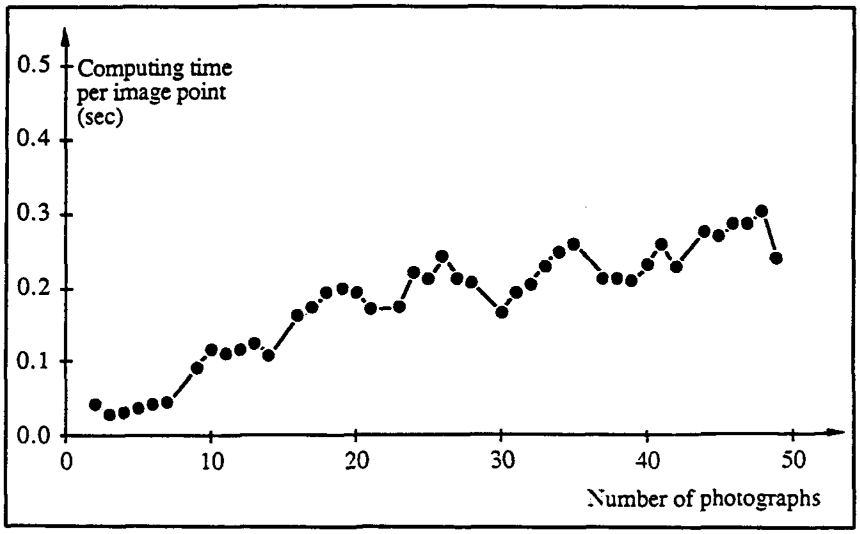

Figure 6: **Block Echallens.** Computing times (solution vector) per additional image point measurement when introducing groups of measurements, using Givens transformations algorithm on a SUN 3 computer upgrated with a 68020 CPU with floating point accelerator

Further information about this test of the performance of GT can be found in the second Interim Report [Holm, 1989b].

An introduction of GT describes Blais (1983):

> *"The Givens Transformations provide a direct method for solving linear least-squares problems without forming the normal equations. One most important feature of this approach for recursive applications is that the solution can be obtained at any stage of the processing of the observational information by simply carrying out a back substitution for the unknown parameter into an upper triangular system of equations."*

Practically, the Givens method is an orthogonal transformation technique based on the use of two-dimensional rotations to eliminate matrix elements. This approach is a special case of the QR decomposition and is comparable with the Cholesky method as described in section 5.2.1. The Givens Transformations process one row of the design matrix A at a time and are used for the sequential addition or deletion of observations in an interactive environment. As mentioned, the solution vector is available at any stage by simply performing a back substitution in an upper triangular equation system. The aforementioned orthogonal decomposition of the design matrix A yields $Q A = R$. Herein, the upper triangular matrix R is identical with the result of the Cholesky decomposition of the normal equation system

A^TPA. When A is a m x n matrix, Q can be interpreted as a sequence of m x m orthogonal matrices, each containing transformation coefficients of one single GT.

Runge (1987) illustrates the functioning of the GT algorithm using an example of a 3 x 3 photo block. In this example, how an additional measurement of a new image point yields a pair of new row vectors containing the partial derivations of the new measurement and the respective constant values is demonstrated. This new observation - two image coordinates in two images - builds the four rows of the new row vector. Individual transformations of these four rows into zero cause alterations in components of the R matrix. Furthermore, a transformation of the respective constant values yields the root of the residual sum of squares. After the transformations the whole new row vector contains zeros and can be filled by the next observation. Similarly, false observations can be removed from the reduced normals, the constant vector, and the square root of the residual sum of squares by GT. When the transformation coefficients have been computed with the diagonal element of j-th row of matrix R and a_j of the design matrix, the elements of matrix R and A will be transformed according to the removed observation. Generally, after updating or "downdating" the reduced normals with GT, the solution vector can be computed with back substitution.

Detailed descriptions and also mathematical formulations of the sequential performance of GT are given in Gruen (1985), Runge (1987), Holm (1989c) and Edmundson (1991).

5.3. Quality control

The major goal of the on-line triangulation technique is to control the quality of the observed data at an early stage of data processing. This permits, in the presence of detected errors, direct re-measurement of observations and provides a cleaned data set for the final execution of the simultaneous block adjustment. In this context, quality control consists of two tasks: (a) detection of blunders, using Baarda's data snooping technique [Baarda, 1968]. (b) compensation of the systematic errors, using additional parameters for self calibration.

5.3.1. Blunder detection and location

Within the least squares estimation approach, Baarda's data snooping technique has proven to be an effective method for blunder detection. This method has been suggested for on-line triangulation by Foerstner (1979), Kratky (1980b), Molenaar (1981c), Dowideit (1982) and Gruen (1982). The method requires only the diagonal elements of the Q_{vv} matrix (weight coefficient matrix of the residuals)

$$Q_{vv} = Q_{ll} - AQ_{xx}A^T \tag{11}$$

with:

Q_{ll}.................. weight coefficient matrix of the observations,
Q_{xx}................ weight coefficient matrix of the unknowns and
A .... design matrix.

A remarkable computational speed-up of the data snooping technique can be achieved with the method of "unit observation vector" [Gruen, 1982], if only a few observations have to be tested at any given time and only the related diagonal elements of the Q_{vv} matrix are required. This is likely to be the case in on-line triangulation, where observations acquired at earlier stages of the sequential process have already passed quality control tests.

Baarda suggested the test criterion w_i

$$w_i = \frac{|v_i|}{\hat{\sigma}_{v_i}} \tag{12}$$

with:

$$\hat{\sigma}^2_{v_i} = \hat{\sigma}^2_o q_{v_i v_i}$$

where:

w_i .... standardized residual,
$\hat{\sigma}^2_o$ .... estimated variance of unit weight and
v_i .... residual,
$\hat{\sigma}^2_{v_i}$ .... estimated variance of residuals,
$q_{v_i v_i}$ .... i^{th} diagonal element of Q_{VV}-matrix.

Under the null-hypothesis $H_{0i} : E(v_i) = 0$, w_i is normally distributed, so that critical values from the normal table can be used for testing. In practice, however, the expected value of the variance of unit weight $\hat{\sigma}^2_o$ is not available. Thus, an estimate of the variance of unit weight $\hat{\sigma}^2_o$ from the simultaneous adjustment or from any sequential stage must be used for testing standardized residuals.

Under H_{0i}, w_i now has a student-t distribution and again critical values for testing are easily available from the respective tables. For example, at a $\alpha = 0.001$ confidence level, (giving a probability of 99.9%), and assuming an infinite number of observations, the critical test value is $c = 3.291$ for the detection of a gross error.

Note that it is assumed that only one blunder is present in the photogrammetric system. But practical experience shows, however, that such an assumption is not realistic. In addition to this problem, correlations between the residuals cause the effects of blunders to be either compensated or accumulated, making blunder detection and location a much more complicated affair.

To perform reliable blunder detection, it is necessary to add more than the traditional two photos of a stereo model into the normal equation system. For higher redundancy and better reliability, a larger number of rays to each object point is required. Thus, more than two photos should be treated as an unit, i.e. computed as a subblock. For an object point, measured with two rays, a blunder in the x-image coordinate is not detectable and in y-image coordinate not locatable. A blunder in the x-measurement of a three ray object point is also not locatable. Consequently, reliable and efficient blunder detection and location requires more than three rays to an object point. According to Gruen (1982), if the overlap is 60%, a subblock of 3 x 3 photos gives optimal local reliability in aerotriangulation.

Practically, the critical aspect from a hardware perspective is the speed/performance of the computation of a test criterion w_i for each observation. Therefore investigations into the computation of the test criteria have been done under the same conditions as mentioned in 5.2.1. Figure 8 shows average computing times for w_i per image point with respect to block size. For this purpose, the test block Echallens was used as a complete block (7 x 7 photos) and two subblock modes (3 x 3 and 5 x 5 photos). In each case, a variety of image point observations, ranging from 2 to 1736 observations, was tested using the data snooping test

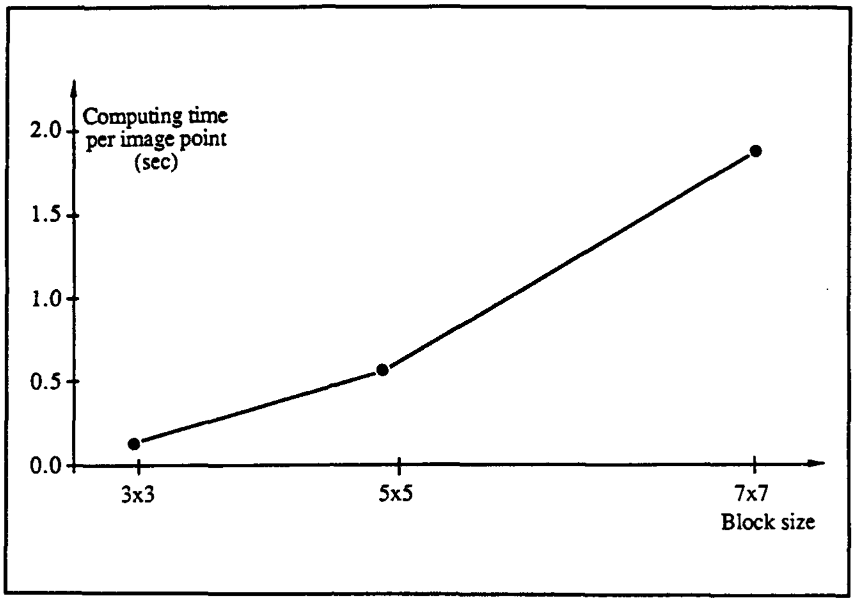

Figure 7: **Block Echallens.** Computing times for w_i per image point with respect to block size on a SUN 3 computer (68020 CPU with floating point accelerator)

criteria w_i. It can be seen that the increase of computing time is nonlinear with respect to an increase in the block size.

5.3.2. Systematic error compensation

The presence of systematic image errors in photogrammetric blocks is a serious problem. Systematic errors can be caused by the following effects:

- Atmospheric refraction
- Geometric distortion by imperfections of lenses
- Change of camera constant and principle point by variation of the temperature and pressure during photographing
- Imprecise fiducial marks
- Vacuum plate unflatness
- Film and stratum deformation
- Blur effected by image motion

Different approaches can be employed to eliminate and/or compensate these systematic errors. The most efficient method is the simultaneous compensation of the systematic errors by expanding the estimation model with additional parameters. This procedure is known as "self calibration". Self calibration has proved its potential in off-line triangulation. If on-line triangulation is performed at a high accuracy level, self calibration is also indispensable.

Two sets of parameters are suggested for the application of aerotriangulation:

- 12-parameter set of Ebner (1976), an "orthogonal model" referring to 3 x 3 image point distribution
- 44-parameter set of Gruen (1978), an "orthogonal model" referring to 5 x 5 image point distribution

The parameter set of Ebner has been used for cartographic and topographic aerotriangulation, while Gruen's parameter set is relevant for applications in cadastral surveying, etc.

In the initial stages of a sequential OLT system the probability of carrying non-determinable additional parameters is certainly higher than in an off-line system. Non-determinable additional parameters can significantly weaken the system and, in extreme cases, lead to near-singularity. In order to avoid running into too many non-determinable additional parameters and spending too much time with the clearance procedure, it is advisable to operate an OLT with a fairly small set of additional parameters, considering only the most important ones which are known to be determinable in those relevant arrangements. A computational fast and reliable procedure for the handling of additional parameters (checking on determinability and possibly significance) has still to be adopted for OLT. Such a procedure has been suggested by Gruen (1978a, 1983a), Foerstner (1981) and Jacobsen (1982). The summarized mathematical formulation of different sets of additional parameters has been published in Kilpelä (1980).

Although preparatory studies for self calibration have been made on this project, due to the redirection of the other project work and the resulting time constraints, its implementation into OLTRIS could not be completed.

6. Software modules of OLTRIS

The OLT system is decomposed into several modules. Figure 8 gives an overview of the main modules with their interrelations and dependencies. The shape of the boxes in the diagrams denotes whether the modules contain mainly functions (rectangles) or are primarily used for data storage (rounded). The following modules contain the administration of OLTRIS (top level):

- USER_INT/XVIEW: User interface handles the communication between the user and the system. The graphical user interface is based on OpenWindows/OPEN LOOK. Within this user interface commands are to be given by menus, buttons, and text in windows.
- FLOW_CON: This administration module contains the flow control of the system for the non graphical user interface. The flow control for the program administrated by the graphical user interface is handled in the module XVIEW.

The core functions are contained in the second level of modules.

- INPUT: The functions of this module will control the data input from outside of the system, organize and store it in internal data structures. Data may come from the user dialogue, measurements (in the final version) or from files.
- INITIAL: The computation of initial values for all unknown parameters is necessary because of the non-linear equation system. Different functions have been implemented for computing the initial values of object points and exterior orientation parameters.
- ADJUST: This adjustment module contains functions for the simultaneous adjustment

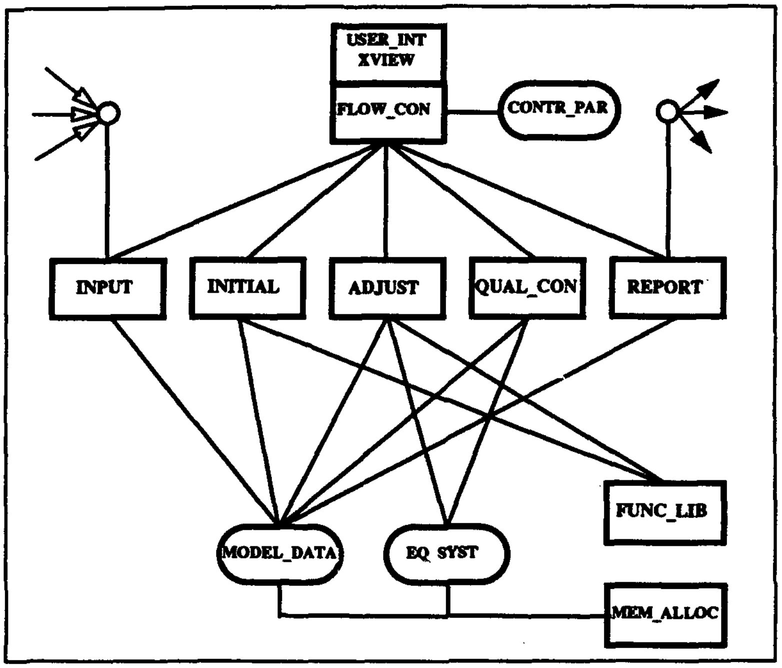

Figure 8: Software modules of the OLTRIS.

(Cholesky factorization and back substitution) and sequential estimation (Givens transformations).

- QUAL_CON: The quality control module contains functions for computing some general statistics, summarizing the results as a whole, performing the blunder detection with data-snooping. It should be expanded with functions for checking determinability and significance of additional parameters.
- REPORT: Functions for the output of results and data as hard copies or on secondary storage for further use are found in this module.

The following two modules contain a library of common functions:

- FUNC_LIB: This module contains a collection (library) for common functions used by different modules. In particular, the functions are for computing mathematical applications, e.g. rotation matrix, design matrix coefficients, and for administrating common applications, e.g. input of values, file handling etc.
- MEM_ALLOC: Functions for memory allocation and reallocation of specific items of the data structures defined in the system are collected in this module.

The following modules contain mainly data:

- CONTR_PAR: This module contains control flow parameters, as well as critical values and other general parameters needed by some computations.
- MODEL_DATA: Data for entities in the mathematical model, e.g. cameras, images, object points, also a priori and computed parameters, observations and design matrix,

are included in this module.

- EQ_SYST: This module includes data and functions for handling the normal equation system and related data.

Detailed description about the modules and its containing functions is given in the software documentation [Holm, Kersten, 1991].

7. The program OLTRIS

OLTRIS has been designed as an on-line data processing system for measuring image point coordinates and adjusting the observations sequentially and/or simultaneously. Additionally, object point coordinates and also exterior orientation elements can be included as observations. Image point coordinates can be measured in mono or stereo mode. Furthermore, the system provides blunder detection with data snooping after the adjustment. Finally, output functionallity (e.g. displaying of information, plotting and saving of data and of results) is provided in OLTRIS.

7.1. The user interface

The program is to be processed by menus, buttons, and text in windows of a graphical user interface. The user interface is based on OpenWindows/OPEN LOOK. A prevoius keyboard/terminal version of the program results from the first implementations of the source code, and was used mainly for testing purposes. An example for keyboard dialog of the program was included as an appendix in the 6th Interim Report. The graphical user interface has been developed with the OpenWindows Developer's Guide (Devguide) 2.0, with more additions under version 3.0. The following discussion concentrates on the use of the program using this graphical interface.

The data processing in OLTRIS can be separated in four phases: initialization, measurement and data processing, quality control, and documentation. The last three phases operate in parallel.

7.2. Initialization

The first phase (initialization) contains the preparation of OLTRIS for on-line data processing, which means, in general, the input of data and the setting of control parameters. In particular, OLTRIS provides for the input of control parameters (e.g. iteration stop criterion, maximum number of iterations etc.), camera parameters (e.g camera calibration data and related informations), the measured image coordinate observations, object point coordinates and exterior orientation elements, using the last two mentioned input options as observations or initial values. The options for initialization - its panel can be started from the on-line triangulation processing panel (see Figure A of Appendix A) - are presented in Figure B of Appendix B. In initialization, the observations of image point coordinates, object point coordinates and exterior orientation elements are read from file and are placed in temporary data structures of the system (internal buffer). During the measurement procedure the observation data are taken from this internal buffer and placed in the data structures of the normal equation system. To date, no direct input from a digital workstation or an analytical plotter is not yet possible. The input of camera calibration data (see Figure D in Appendix C) and image or model data (see Figure E in Appendix C) is possible within the initialization menu and also later within the input option of the on-line triangulation processing panel. Figure 9 gives an overview of the data flow in the system. The left side of the

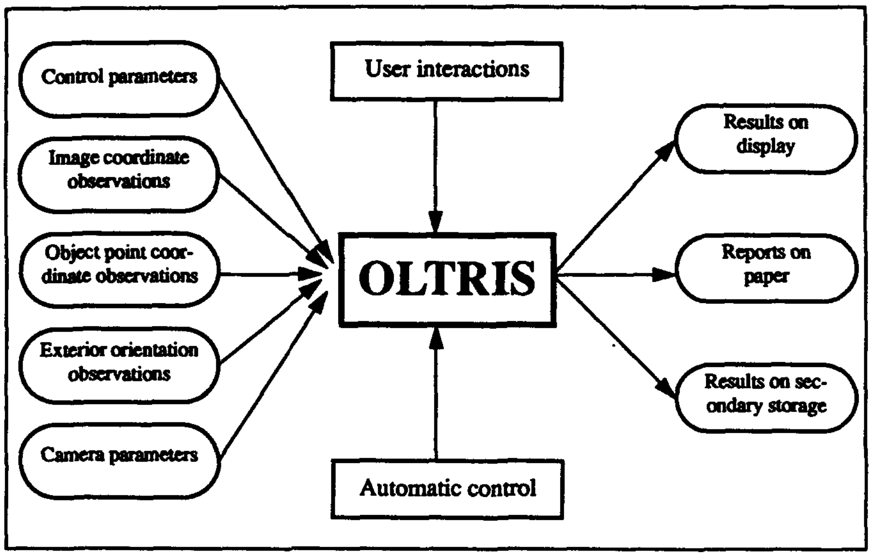

Figure 9: OLTRIS and its data flow

figure shows the input possibilities into the system, while on the right the output of results and data is presented. Between processing (as far as the data are processed), the results and also the processed data can be displayed and plotted on screen, reported on paper and/or stored on disk. Automatic control will be achieved by setting control flow parameters - Figure C in Appendix B illustrates its panel - and the user interactions lead the processing of the system.

7.3. Measurement and data processing

The second phase of data processing in OLTRIS is for image measurement and data processing. When starting up the program, the on-line triangulation processing panel (main panel) appears (see Figure A in Appendix A) on screen, from which all processes can be managed. After initialization of parameters and data input as mentioned before, the measurement procedures can be started from the main menu. The following possibilities for

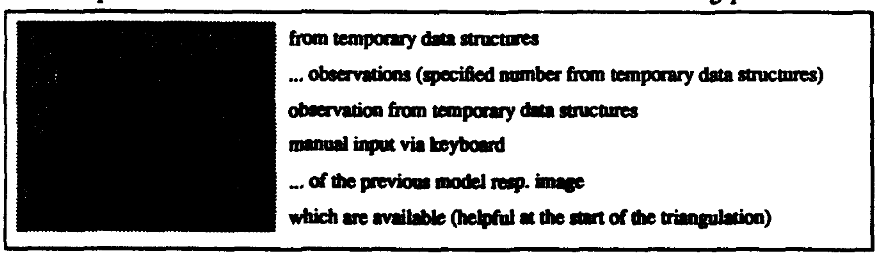

Figure 10: Measurement options of OLTRIS

measuring image point coordinates as depicted in Figure 10 can be selected by the operator.

When beginning a new triangulation it is advisable firstly to measure in the start image (or model) all available control points in order to fix the datum and secondly to continue with "next observation" or "next and more". Continuing triangulation by using the option "tie points" all following images (or models) will be connected in the block sequentially. The measurement options "One specific" and "Manually" provide an interaction of the operator for measuring certain points. The measured points will be plotted on-line in a window (see Figure I in Appendix E) to show the contribution of points in the images. After each measurement the normal equation system is updated sequentially with the new observations, if initial values for the respective object point and exterior orientation parameters of the newly measured image are available. Otherwise the initial values will be computed when displaying data or relinearizing and updating the equation system by pressing the button *Display* or *Update*. The functional co-operation of the sequential and simultaneous relinearization and update of the normal equation system, when introducing new (additional) observations, is illustrated in Figure 11. The strategy and the functions for computing initial values are described in section 5.1. The adjustment procedures are introduced in section 5.2. To control the adjustment the operator has to choose the iteration criterion (average changes of the rotation angles or of the coordinates, or the average changes of v^Tpv), the iteration stop criterion and the maximum number of iterations (in initialization).

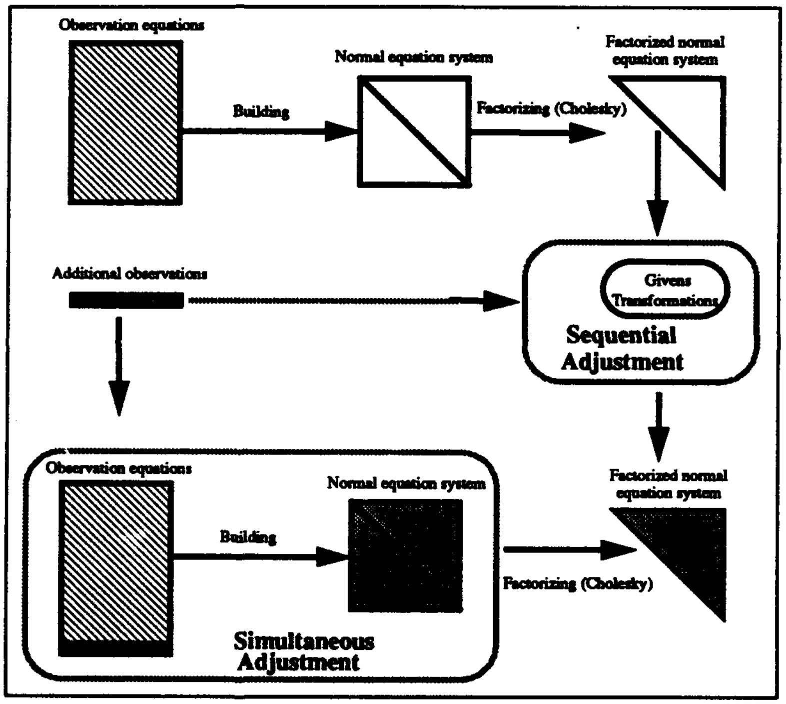

Figure 11: Relinearizing and updating of the normals by Givens Transformations and Cholesky

7.4. Quality control

During measurement and data processing OLTRIS provides a quality control of the observations with data snooping. The goal of the quality control is the detection and location of blunders in the observed data. To this end, in addition to image point coordinate observations, all available observations of exterior orientation elements and object points will be tested by the data snooping method. An example of a detected blunder is shown in Figure F (Appendix D). For cleaning the data set, a detected blunder can be **excluded** from the adjustment. It is recommended to **remove** the detected observation finally from the internal buffer only after performing a simultaneous adjustment in order to the see the effect of its exclusion.

7.5. Documentation

During and after measurement the operator may want to see up-to-date results and data records. Therefore, the third phase (documentation) cannot be separated from the measurement and data process phase. The documentation of results and processed data is separated in *Display*, its options are depicted in Figure 12, and in *Report*, its options are *Save, Print* and *Plot*. The panels of these report options are shown in Appendix D, Figures F,G,H.

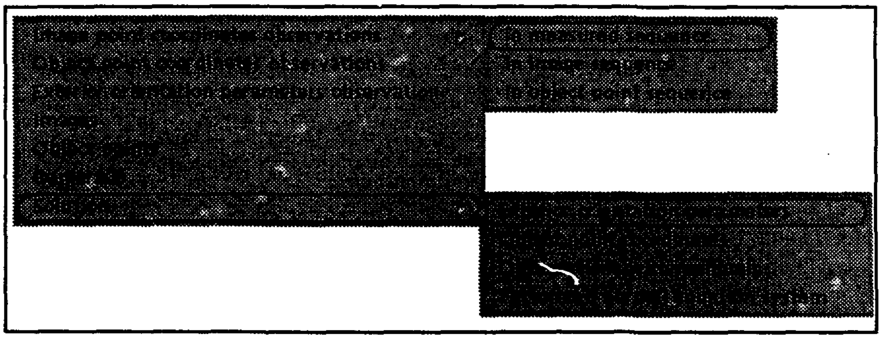

Figure 12: Display options of OLTRIS

7.6. Results

To illustrate the performance and functionary of OLTRIS, a step-by-step triangulation of a photogrammetric block is demonstrated stepwise. The data of the photogrammetric block Simplon, a mountainous area, has been chosen for the demonstration. In order to reduce the quantity of shown data and results to a minimum in the report, only the start and the end of the triangulation will be presented. The chosen block was flown in four strips with 60% overlap in strip and cross-strip direction. The processing of the block (4 x 6 images, 102 object points, 943 observations, 450 unknowns and 493 redundant observations) was executed on a Sun SparcStation 1. The first seven exterior orientation elements are introduced as observations thereby fixing the datum of the photogrammetric block. After introducing the two first images and measuring 6 points in each image (see Figure J in Appendix E), at this stage, the adjustment can be computed simultaneously for efficiency and stability reasons. Having one redundant observation (see Figure K in Appendix E), the sigma a posteriori has been estimated in the adjustment as 11.7 microns, compared to sigma a priori defined at 5.0 microns. Adding more images and observations increases the redundancy and the photogrammetric block becomes more stable. The performance of the sequential

update of the normal equation system when introducing new observations is shown in Figure 7, but processing a different photogrammetric test block on a Sun 3 workstation. The result of the final adjustment of the Simplon block is presented in Figure L, in Appendix F. A confirmation of the quality of the mathematical model is the value of sigma a posteriori, which, as seen, does not differ greatly from the introduced value of sigma a priori. As an example of the results the solution of the exterior orientation elements is depicted in Figure M of Appendix F.

In summary, a first version of the system has been implemented on the Sun workstation. In the future, it has to be decided, to which measurement system - analytical plotter or digital workstation - OLTRIS should be connected. Finally, there is always room for improvement of a very complex program like OLTRIS.

8. Conclusions

To the authors' knowledge, no fully operational OLT system has been realized yet. A realization of a practical OLT system could look as depicted in Figure 13. The left side of the Figure illustrates the input into the system of direct image measurements by an Analytical plotter or of digital images by an image acquisition station via a scanner or via CCD-cameras. The data, as shown on the right side of Figure 13, can be saved on disk or printed as hard copies. Automatic control and user interactions lead the processing of an OLT System.

Theoretical OLT problems, except statistical testing in sequential mode, have been solved, but have not been practically combined into a single system. The continual improvement in computer performance has caused reduced efforts in the development of fast processing algorithms for OLT. Furthermore, investigations into instrument development are concentrating on the improvement of automatic measurement procedures on analytical plotters instead of implementing fast data processing algorithms, including quality control to detect gross errors and to check precision and reliability of the network.

The authors see potential in further development and use of the OLT algorithms and methods for applications in robotics, industrial quality control and autonomous vehicle navigation. Of particular interest in robotics and autonomous vehicle navigation is, for example, the task of determining the position and orientation of a mobile robot on-line. Here on-line data processing could also play an important role.

9. Acknowledgment

The project "On-line point positioning with single frame camera data" is sponsored by the U.S. Government, Department of Defense, represented through its European Research Office of the U.S. Army in London. This support is gratefully acknowledged.

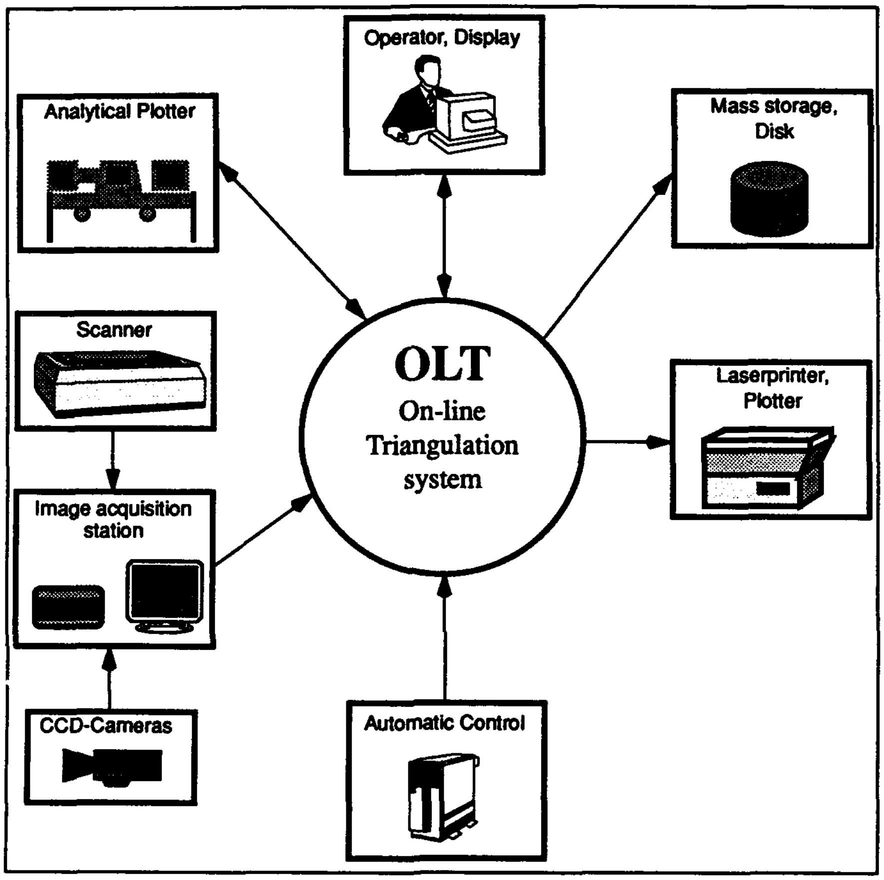

Figure 13: Hardware configuration of an OLT system

10. References

10.1. Internal reports

Kersten, Th., Holm, K. R., 1991. Software Documentation (SD.00 - SD.03), Version 2.0, June 1991, IGP/ETH-Zurich.
Holm, K. R., 1989a. 1st Interim Report (Jan.-Feb. 89), IGP/ETH-Zurich.
Holm, K. R., 1989b, 2nd Interim Report (Mar.-Jun. 89), IGP/ETH-Zurich.
Kersten, Th., 1990a. 3rd Interim Report (Jul.-Dec. 89), IGP/ETH-Zurich.
Kersten, Th., 1990b. 4th Interim Report (Jan.-Jun. 90), IGP/ETH-Zurich.
Kersten, Th., 1991a. 5th Interim Report (Jul.-Dec. 90), IGP/ETH-Zurich.
Kersten, Th., 1991b. 6th Interim Report (Jan.-Jun. 91), IGP/ETH-Zurich.

10.2. General references

Baarda, W., 1968. A testing procedure for use in geodetic networks. Netherlands Geod. Com. Publications on Geodesy. Vol.2, No.5, Delft.

Baarda, W., 1977. Measures of the accuracy of geodetic networks. IAG Symposium on Optimization of Design an Computation of Control Networks, Sopron, pp. 419-436.

Blais, J. A. R.,1983. Linear least-squares computations using Givens transformation. The Canadian Surveyor, Vol. 37, No.4, pp. 225-233.

Cogan, L., 1986. Design concepts of on-line aerial triangulation software. ASP/ACSM Convention, Washington.

Dorrer, E., 1978. A contribution to sequential on-line numerical aerial triangulation. Presented Paper, Symposium of the Commission III of ISP, Moscow.

Dowideit, G. R., 1980. On-line bundle adjustment for analytical plotter. Int. Archives of Photogr., Vol. XXIII, B3, Commission III, pp. 168-177.

Dowideit, G. R.,1982. Experience with improved on-line-bundle-block-adjustment on the APC3. Proceedings of the Symposium of Com. III of ISPRS Helsinki, June 7-11, 1982.

Ebner, H., 1976. Self Calibrating Block Adjustment. Int. Arch. Phot., Vol. XXI, Part 3, Invited Paper, Com. III, ISP Congress, Helsinki.

Edmundson, K. L., 1991. On-line triangulation of sequential stereo-pairs using Givens Transformations without square roots. Master Thesis, Department of Geodetic Science and Surveying, The Ohio State University.

Foerstner, W., 1979. Sequential checking of model coordinates with a program for strip adjustment. Presented Paper, Symposium on On-Line Triangulation, ISP WG III/4 , Ottawa.

Gruen, A., 1978. Progress in photogrammetric point determination by compenstion of systematic errors and detection of gross errors. Presented paper at Symposium of the Commission III of ISP Moscow, 1978.

Gruen, A., 1982. An optimum algorithm for on-line triangulation. Photogrammetric Engineering and Remote Sensing, Vol.51, No.4, pp.419-436.

Gruen, A., 1985. Algorithmic aspects in on-line triangulation. Paper presented to the Symposium of the Commission III of ISPRS Helsinki, June 7-11, 1982.

Gruen, A., 1986a. Photogrammetrische Punktbestimmung mit der Bündelmethode. Institut für Geodäsie und Photogrammetrie an der ETH Zürich, Mitteilungen Nr. 40.

Helmering, R. J., 1977. A general sequential algorithm for photogrammetric on-line processing. Photogramm. Eng. and Remote Sensing, Vol. 43, pp. 469-474.

Holm, K. R., 1989c. Test of algorithm for sequential adjustment in on-line triangulation. Photogrammetria(PRS), 43, pp.143-156.

Kratky, V., 1979. On-line analytical triangulation. The Canadian Surveyor, Vol. 33, No.2.

Kratky, V., 1982. Present status of on-line analytical triangulation. Photogrammetria(PRS), 38, pp.1-12.

Mikhail, E. M., Helmering, R. J., 1973. Recursive methods in photogrammetric data reduction. Photogramm. Eng., 39(9), pp. 983-989.

Runge, A., 1987. The use of Givens Transformation in on-line phototriangulation. Proceedings of Intercommission Conference on Fast Processing of Photogrammetric Data, Interlaken, Switzerland, June 2-4, 1987, pp. 179-203.

Wyatt, A. H., 1982. On-line photogrammetric triangulation - An algorithmic approach. Master Thesis, Department of Geodetic Science and Surveying, The Ohio State University.

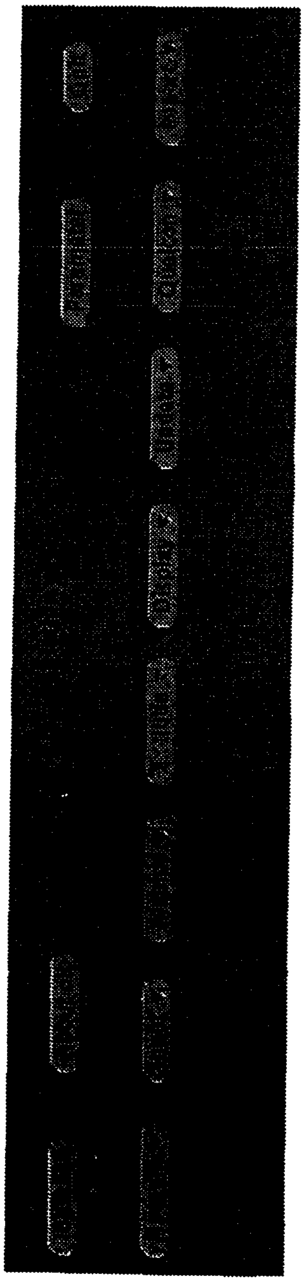

Figure A: The on-line triangulation data processing panel

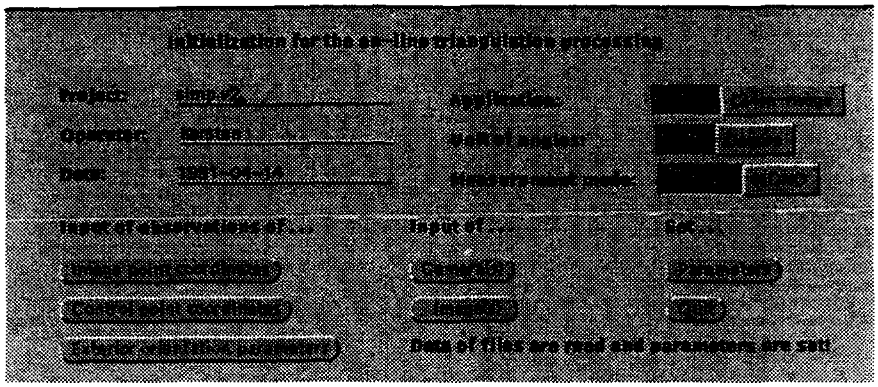

Figure B: The on-line triangulation initialization panel

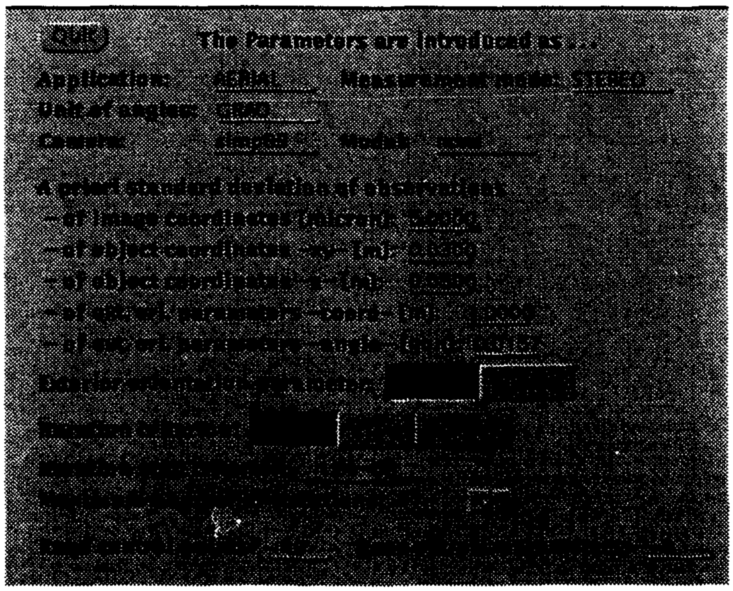

Figure C: Subpanel of the on-line triangulation initialization panel

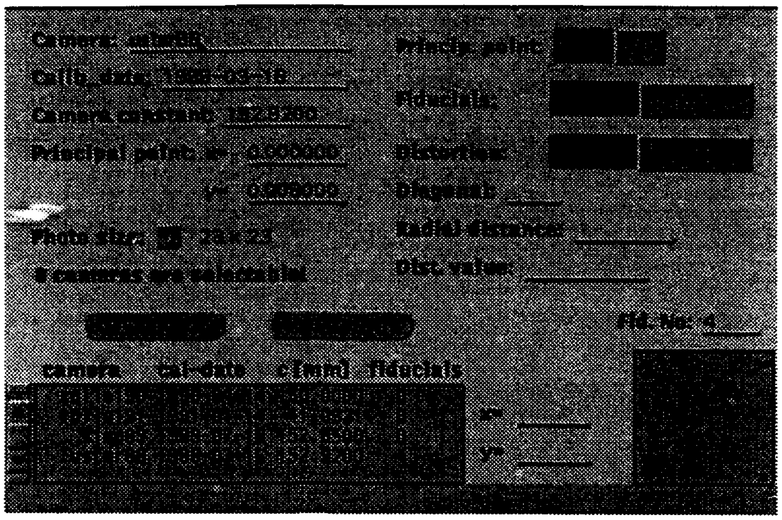

Figure D: Input of camera calibration data (Subpanel of the on-line triangulation initialization panel)

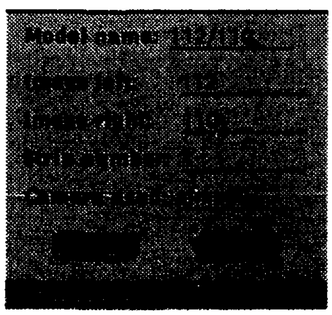

Figure E: Input of image or model data (Subpanel of the on-line triangulation initialization panel)

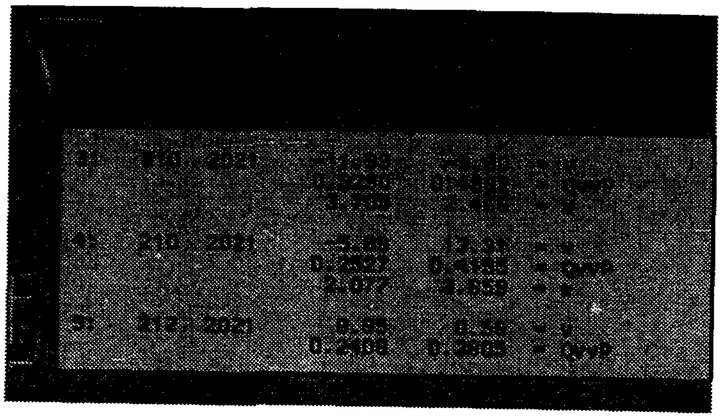

Figure F: Display for quality control: detected blunder in point 2021 of image 210

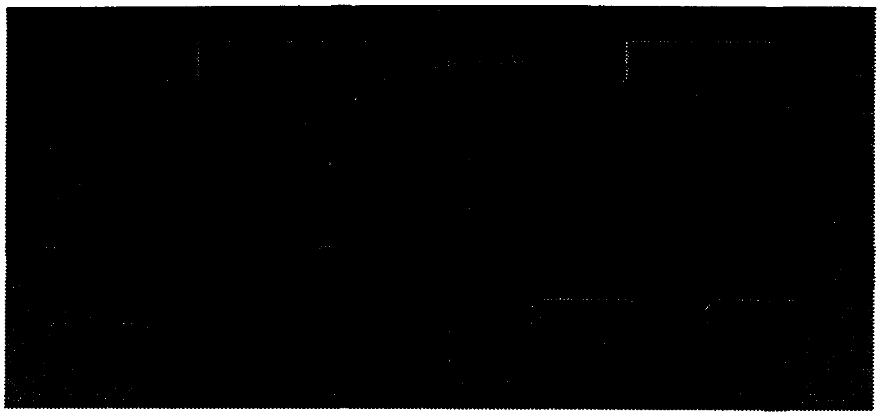

Figure G: Panel for printing data and results

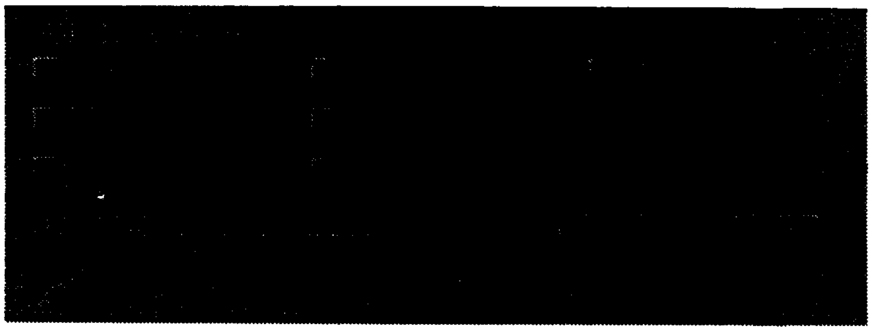

Figure H: Panel for saving data and results

Figure I: Panel for plotting data and results

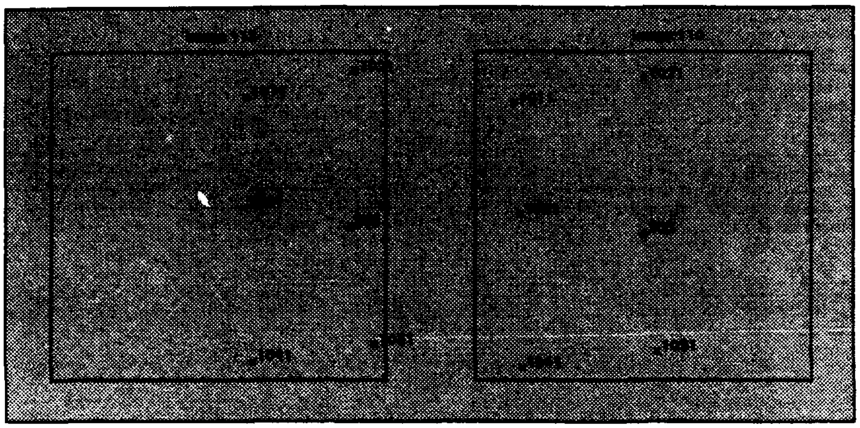

Figure J: Display of the measured points in first two images

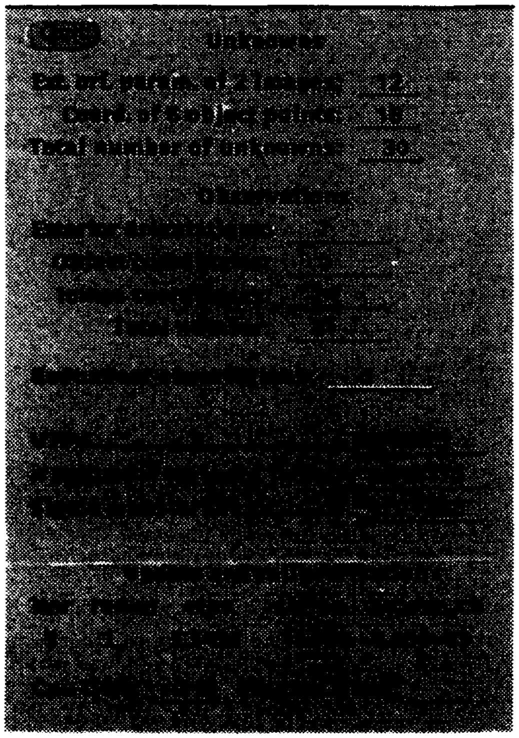

Figure K: Display of triangulation status after the first model

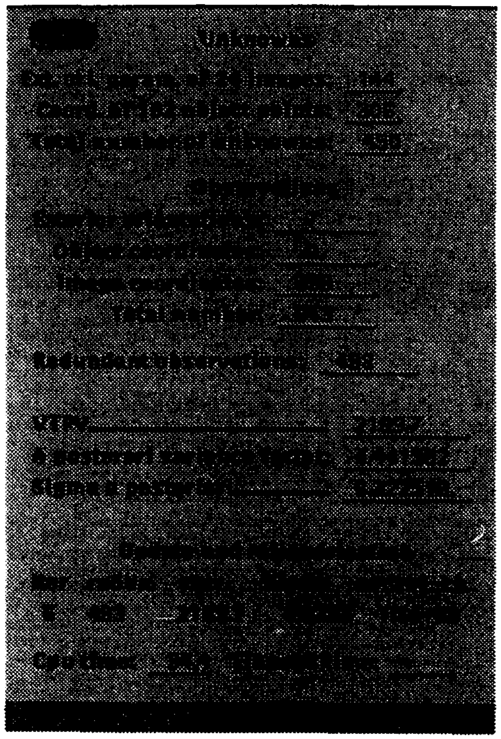

Figure L: Displayed results of final adjustment of the Simplon block

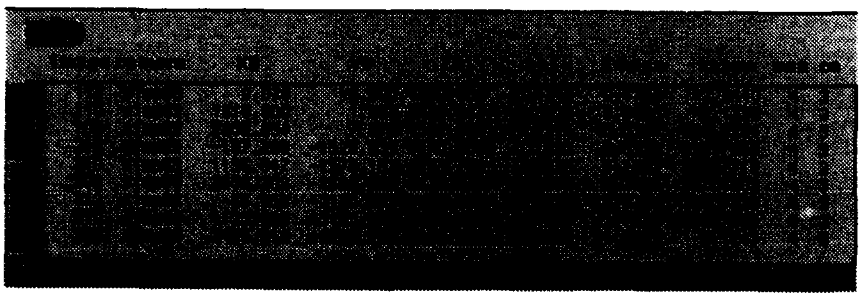

Figure M: Display of results: exterior orientation elements of the images of the photogrammetric block

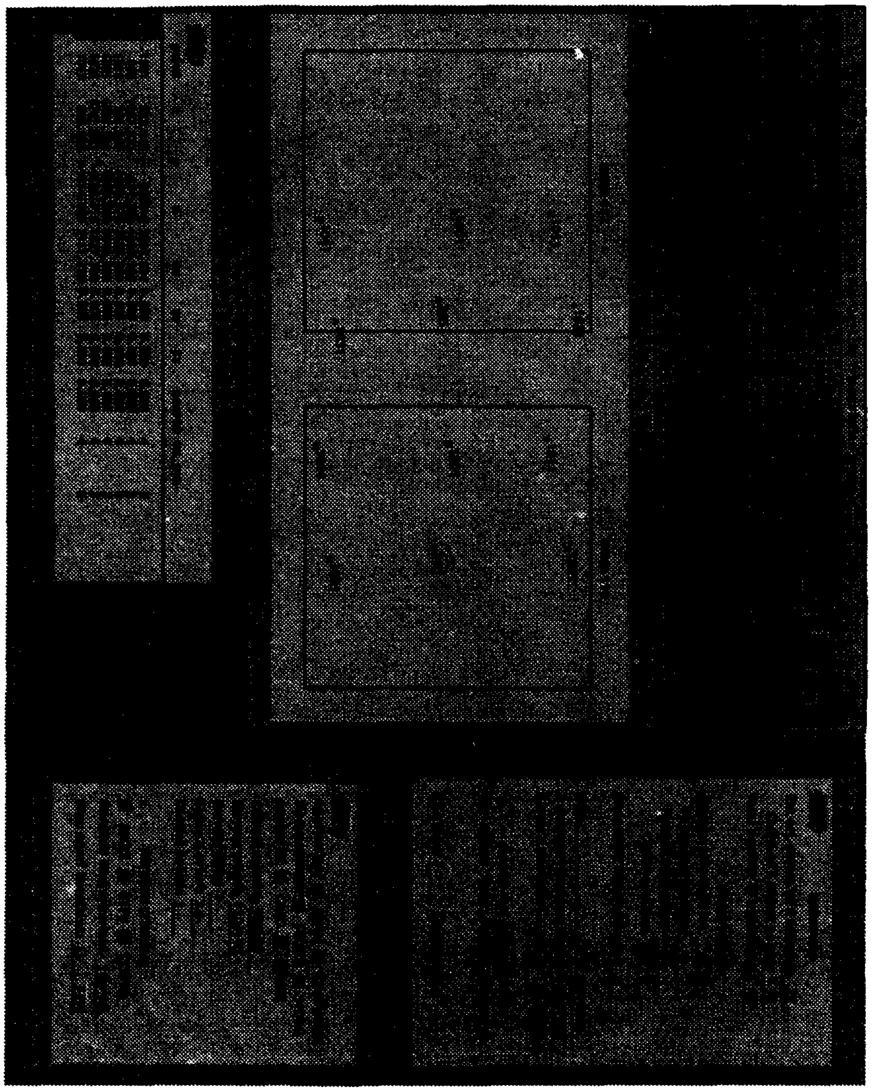

Figure N: Screen dump of the window environment of OLTRIS

SOFTWARE DOCUMENTATION CONTENT

SOFTWARE DOCUMENTATION SCHEME

1. Scope of the documentation

This software documentation is based on the recommendations outlined in IEEE 1016. A rigorous "Software Requirements Specification" as described in IEEE 830 is not made in this project. The requirements for the software are instead embedded in this description.

2. Description Organisation

IEEE 1016 (Section 6.) contains four different methods for describing the design of software. Two of these four approaches, *Decomposition* and *Detailed Design* have been chosen for documenting the software of OLTRIS.

Decomposition: Identification, Type, Purpose, Function, Subordinates Dependecies, Resources.

Detailed Design: Identification, Function, Interface, Processing, Data.

The characteristics of the design description are defined in section 3.

3. Definitions and acronyms used

Design Entity: (See IEEE 1016, Sect. 5.2) "... an element (component) of a design that is structurally and functionally distinct from other elements and that is separately named and referenced". Objective of decomposition is "to divide the system into separate components that can be considered, implemented, changed, and tested with minimal effect on other entities".

Entity Attribute: (See IEEE 1016, Sect. 5.3) "... named characteristic or property of a design entity".

The individual attributes which are addressed in the design descriptions, are listed below, with short(ened) descriptions (see IEEE 1016, 5.3.1-5.3.10):

1. Identification A name; unique; characterizing the nature of the entity.

2. Type Kind of entity. Subprogram, module, procedure, process, data store. In this documentation, "module" means a design entity, without regarding how it will be implemented, while "program module" indicates a physical module e.g. in a C program.

3. Purpose A description of *why* the entity exists - ...the specific functional and performance requirements for which this entity was created.

4. Function A statement of what the entity does - ... the transformation applied to the inputs to produce the output (Details belong in 9. Process). For data entity: Type of information stored or transferred.

5. Subordinates Identification of all entities composing this one.

6. Dependencies Descriptions of the relationships with other entities. Shall identify the "uses..." or "requires the presence of ..." relationship. May be depicted by structure charts, data flow diagrams, transaction diagrams.

7. Interface Description of how other entities interacts with this entity. Methods and rules, including e.g. communicating through parameters and direct access to internal data (Details in IEEE 1016).

8. Resources Description of elements that are external to the design. Physical devices, software services (e.g. libraries), processing resources.

9. Processing	Description of the rules used by the entity to achieve its function. Algorithm, contingencies. A refinement of the Function attribute. Should include: timing, sequencing, prerequisits for initiation, priority of events, ... , termination criteria, handling of contingencies.
10. Data	Description of data elements internal to the entity. Representation, format, structure, initial values, meaning and use, acceptable values. May be in the form of a data dictionary.

4. References

IEEE 1016 — IEEE Std 1016-1987: "IEEE Recommended Practice for Software Design Descriptions".

IEEE 830 — IEEE Std 830-1984: "IEEE Guide to Software Requirements Specifications".

OVERVIEW AND COARSE DECOMPOSITION

OLTRIS - The top module of the system

1. Identification OLTRIS

2. Type Module.

3. Purpose This module is created to give an overview over the total system and its environment, as briefly outlined in Figure 1.

The system is part of a research project with the aim to "... develop a data processing concept for **on-line high accuracy point positioning ... systems**". To find the best solutions for this concept, there are pointed out four research topics. It is decided that the concept shall be materialized in an **operational system** for use in **aerial photo-triangulation**. The system should work on **digital images**.

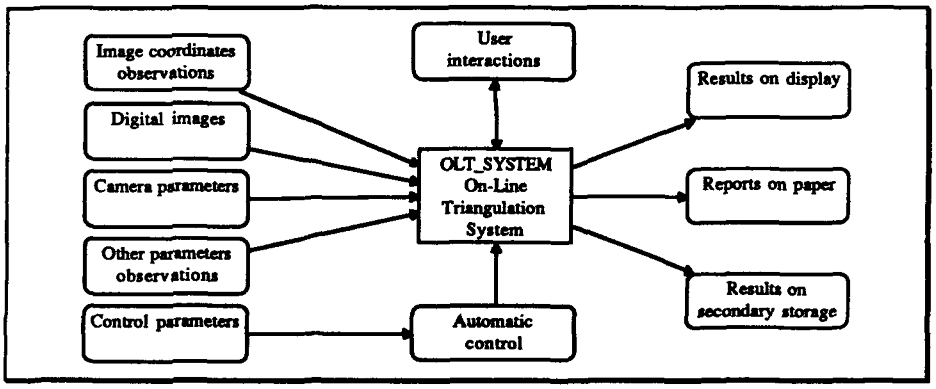

Figure 1: The On-Line Triangulation System OLTRIS and its environment.

4. Function The system will perform On-Line Photo-triangulation. Important tasks are:

- Initial value computation
- Sequential and simultaneous bundle adjustment
- Blunder detection
- Self calibration, with test of significance and determinability

(• Automatic point transfer)
(• Inclusion of kinematic GPS data)

Currently the program "oltris" is working as an experimental program without a link to analytical plotter or digital workstation.

5. Subordinates (See Figure 2) This module is composed of following modules:

USER_INT/XVIEW	- User interface in OpenWindows (OpenLook)
FLOW_CON	- Administration module
INPUT	- Data input
INITIAL	- Initial parameter values computation
ADJUST	- Bundle adjustment
QUAL_CON	- Quality control
REPORT	- Output of results
FUNC_LIB	- Library for common functions
MEM_ALLOC	- Memory allocation functions
CONTR_PAR	- Control parameters
MODEL_DATA	- Data for entities in the mathematical model
EQ_SYST	- Normal equation system and related data

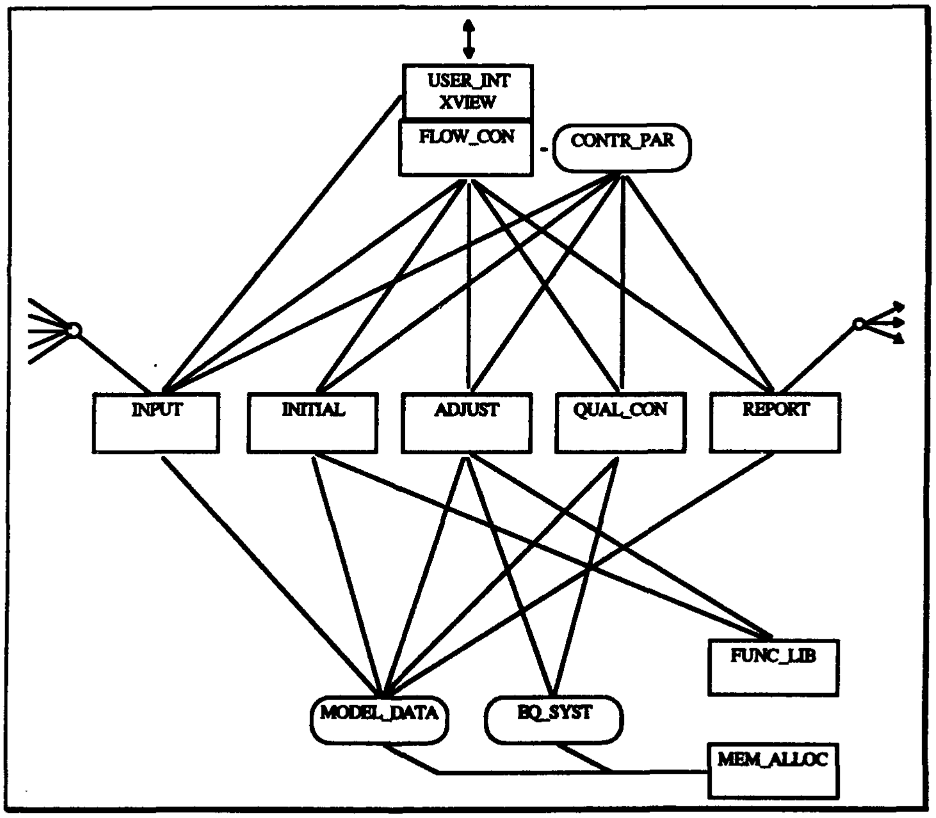

Figure 2: An overview of the main modules of the on-line triangulation system. The shape of the boxes denotes whether the module contains mainly functions (right-angled rectangles) or mainly data storage (rounded)

6. Dependencies A session with this system is supposed to be conducted interactively by a human operator.
Requiring its respective source equipment, data input is supposed to be any or more of: Digital images, coordinate lists, parameter lists, control commands. Input may occur manually or from disk, depending on the character of the data.
Output of results is sent to disk, printer, and display.

8. Resources The **digital on-line photo-triangulation system** is to be implemented in a **UNIX** environment on a **Sun** workstation, using the programming language "C". The user dialog will be handled through a window system (OpenWindows).

The capacity of the system should be practically unlimited concerning number of image points, object points, photographs, and cameras. Response times should be well within the limits of of the operator's patience.

Both response time and memory requirements are to be more clearly specified.

As examples of possible memory requirements:
• For normal equation system and observation data, 50 photos and 500 points, each measured in 4 photos, may need 2 Mb, while 100 photos and 1000 points need 5 Mb. This should stay resident in memory during sequential adjustment.
• For digital images (or parts of images) available for on-screen viewing, 10 image patches of 512x512 8-bits pixels will occupy 2.5 Mb. However, only a few need be in memory at one time.
• For digitized points, which are defined by small patches (≤ 25x25 pixels), 1000 image points may require about 1/2 Mb. And again, not all are needed in memory at one time.

DECOMPOSITION DESCRIPTION

Table of contents

1. Introduction

This document describes the decomposition of the On-Line Triangulation System (OLTRIS), also showing the dependencies between the modules. Figure 1 gives an overview over the main modules with their interrelations. The shape of the boxes in the diagrams denotes whether the modules contain mainly functions (rectangles) or primarily used for data storage (rounded).

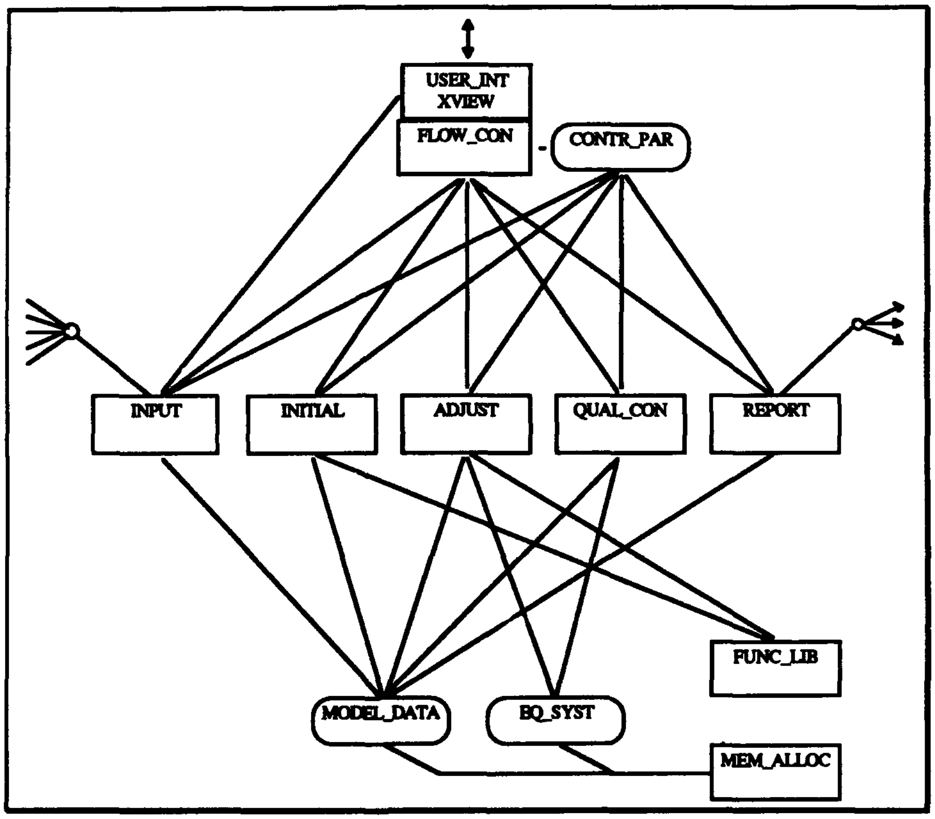

Figure 1: An overview of the main modules of the OLTRIS.

2. USER_INT/XVIEW - User interface in OpenWindows

1. Identification USER_INT/XVIEW

2. Type Module.

3. Purpose Handle all dialogs between the user and the system, by which the user can lead the process of the system. Commands are to be given by menus, buttons, and text in windows, - whatever is found appropriate for the specific case. The dialogs include also any manual input of parameters.

4. Function In a OpenWindow environment it seems natural to include all "set-up" activity for the interactive tools in this module, while action-taking functions ("Target Functions") belong in FLOW_CON.

This module should also contain functions needed for communicating with OpenWindows when the working modules need to change or add elements of the user interface.

5. Subordinates This module contains the following modules:

UI_DATA	- User interface for all data
UI_SYST	- User interface for the system
UI_CANV	- User interface for the canvases
XV_NOTEV	- Notify and event callback functions

6. Dependencies (See Figure 1)

- Part of module OLTRIS
- Interacts with module

  FLOW_CON

  and (possibly indirectly) with

  INPUT
  INITIAL

8. Resources Hardware: Sun workstations

Software: OpenWindows Version 3.0 (OpenLook)

Libraries:
- libxview.a
- libolgx.a
- libX.a

2.1. UI_DATA - User interface for the data

1. Identification UI_DATA

2. Type Module.

3. Purpose Provide panels, windows etc for tasks of data handling.

4. Function Contains the UI object initialization functions for the window applications of data input, displaying etc. For this reason the objects of XView data types (popup frame, menus, buttons, textpanels, etc.) will be created.

5. Subordinates This module contains the following functions:

CAMERA_UI	- Window for camera data input and display
IMAGE_UI	- Window for image data input and display
DISPLAY_UI	- Window for for display of data and results
INOBS_UI	- Window for input of observations

6. Dependencies • Part of module USER_INT/XVIEW

• Interacts with the module

FLOW_CON

8. Resources Software: OpenWindows Version 3.0 (OpenLook)

Libraries: - libxview.a, libolgx.a, libX.a

2.2. UI_SYST - User interface for the system

1. Identification UI_SYST

2. Type Module.

3. Purpose Provides windows for processing of OLTRIS.

4. Function Contains the UI object initialization functions for the window applications of the system processing etc. For this reason the objects of XView data types (base and popup frame, canvase, menus, buttons, textpanels, etc.) will be created.

5. Subordinates This module contains the following functions (windows):

OLTINI_UI	- Dialog for program flow control
OLTMAIN_UI	- Dialog for adjustment control parameter
PARASET_UI	- Dialog for initial value control parameter
GENSTAT_UI	- Dialog for program flow control
PINFO_UI	- Dialog for program flow control
QUALCON_UI	- Dialog for adjustment control parameter
REPORT_UI	- Dialog for initial value control parameter

6. Dependencies • Part of module USER_INT

• Interacts with the module

FLOW_CON

8. Resources Software: OpenWindows Version 3.0 (OpenLook)

Libraries: - libxview.a, libolgx.a, libX.a

2.3. UI_CANV - User interface for the canvas applications

1. Identification UI_CANV

2. Type Module.

3. Purpose Provide canvases for OLTRIS for displaying positions of measured image points and images in the photogrammetric block.

For image coordinates: In the final system, there should be one or more canvases for displaying (parts of) digital images. Positioning on the images is to be done by mouse and cursor. (Image matching techniques will also be used for precise positioning, but therefore an interface must be provided to link the olt system with a template matching program.)

4. Function Contains the UI object initialization functions for the canvas applications of the system. For this reason the objects of XView data types (canvase, menus, buttons, etc.) will be created.

5. Subordinates This module contains the following functions (canvases):

CANV_BLOCK_UI - Canvas for the photogrammetric block.
CANV_IMG_UI - Canvas for display of images and meas. points

6. Dependencies •Part of module USER_INT

• Interacts with the module

INPUT

8. Resources Software: OpenWindows Version 3.0 (OpenLook)

Library: - libxview.a, libolgx.a, libX.a

2.4. XV_NOTEV - Notify and event callback functions

1. Identification XV_NOTEV

2. Type Module.

3. Purpose Provide notify and event callback functions.

4. Function Window system: Set up the user dialog for object parameter input.

5. Subordinates This module is composed of the functions:

DL_IMAGE - Dialog for image data input
DL_CAMERA - Dialog for camera data input
DL_MCAM - Dialog for manual camera data input
DL_OPC - Dialog for object point coordinates input
DL_EOP - Dialog for exterior orientation parameters input

6. Dependencies • Part of module USER_INT

• Interacts with the module

INPUT

8. Resources Software: OpenWindows Version 3.0 (OpenLook)

Libraries: - libxview.a, libolgx.a, libX.a

3. FLOW_CON - Administration module

1. Identification FLOW_CON

2. Type Module.

3. Purpose To perform the central control of the system, based on user commands and intermediate results in the system.

4. Function This module contains the main program and functions which, based on user commands and intermediate results, will administer the control flow. All calls to main functions in the "working" modules of the system should then come from this module.

When using a window system, this module includes the "Target Functions" activated by interactive tools (OpenWindows) to interpret the events and decide actions to be taken.

5. Subordinates This module contains the following modules:

FC_MAIN	- Top-level flow control
FC_MEASURE	- Interface to the INPUT module (Measurement)
FC_INPUT	- Interface to the INPUT module
FC_INCLUDE	- Inclusion of observations
FC_EXCLUDE	- Exclusion of observations
FC_DISPLAY	- Display of data and results on screen
FC_UPDATE	- Interface to the ADJUST module
FC_QUALCON	- Interface to the QUAL_CON module
FC_REPORT	- Interface to the REPORT module

6. Dependencies (See Figure 1)

- Is part of OLTRIS
- Supposed to interact with all the main modules composing OLTRIS

8. Resources None required.

3.1. FC_MAIN - Top-level flow control

1. Identification FC_MAIN

2. Type Module.

3. Purpose Provide the central administration of OLTRIS.

4. Function This module performs the initialization of XView, of the user interface and OLTRIS components

5. Subordinates This module contains the function:

XOLT - Top-level flow control

6. Dependencies
- Is part of module FLOW_CON
- Interacts with the modules

USER_INT/XVIEW

FC_MEASURE
FC_INPUT
FC_INCLUDE
FC_EXCLUDE
FC_DISPLAY
FC_ADJUST
FC_QUALCON
FC_REPORT

8. Resources None.

3.2. FC_MEASURE - Interface to the INPUT module

1. Identification FC_MEASURE

2. Type Module.

3. Purpose Provides an interface to the module INPUT (Measurement).

4. Function This module performs detailed control of the operations in the INPUT module with respect to input of image point coordinates.

5. Subordinates This module contains the function:

NEXT_OIPC	- Input of next image point observations
MORE_OIPC	- Input of more than one image point observations
SPEC_OIPC	- Input of a specific image point observations
MAN_OIPC	- Manual input of image point observations
TIE_OIPC	- Input of image point observations of tie points
CONPO_OIPC	- Input of image point observations of control points

6. Dependencies • Is part of module FLOW_CON

• Interacts with the modules

UI_SYST
INPUT

8. Resources None required.

3.3. FC_INPUT - Interface to the INPUT module

1. Identification FC_INPUT

2. Type Module.

3. Purpose Provides an interface to the module INPUT.

4. Function This module performs detailed control of the operations in the INPUT module.

5. Subordinates This module contains the following functions:

XV_CAMERA	- Input of camera calibration data
XV_IMAGE	- Input image data
XV_OOPC	- Input of object point observations
XV_OEOP	- Input of exterior orientation parameters

6. Dependencies • Is part of module FLOW_CON

• Interacts with the modules

UI_DATA
UI_SYST
INPUT

8. Resources None required.

3.4. FC_INCLUDE - Inclusion of observations

1. Identification FC_INLUDE

2. Type Module.

3. Purpose Includes (additional) observations in the adjustment.

4. Function This module includes additional observations in the adjustment.

5. Subordinates This module contains the functions:

ALL_INCL_OIPC - Include of all unused ipc observations
ALL_INCL_OOPC - Include of all unsused opc observations
ALL_INCL_OEOP - Include of all unsused eop observations
SPEC_INCL_OIPC - Include of one specific unused ipc observations
SPEC_INCL_OOPC - Include of one specific unused opc obs.
SPEC_INCL_OEOP - Include of one specific unused eop obs.

6. Dependencies • Is part of module FLOW_CON

• Interacts with the modules

INPUT

8. Resources None required.

3.5. FC_EXCLUDE - Exclusion of observations

1. Identification FC_EXCLUDE

2. Type Module.

3. Purpose Excludes observations from the adjustment.

4. Function This module excludes observations from the adjustment.

5. Subordinates This module contains the functions:

SPEC_EXCL_OIPC - Exclude of one specific unused ipc obs.
SPEC_EXCL_OOPC - Exclude of one specific unused opc obs.
SPEC_EXCL_OEOP - Exclude of one specific unused eop obs.

6. Dependencies • Is part of module FLOW_CON

• Interacts with the modules

QUAL_CON

8. Resources None required.

3.6. FC_DISPLAY - Display of data and results on screen

1. Identification FC_DISPLAY

2. Type Module.

3. Purpose Provide an interface to the module REPORT.

4. Function This module performs detailed control of the operations in displaying data and results on screen.

5. Subordinates This module is composed of the modules:

XV_OIPC	- Display of ipc observations in measured sequence
XV_OIPC_IMG	- Display of ipc observations in image sequence
XV_OIPC_OPT	- Display of ipc observations in object point sequence
XV_OOPC	- Display of object point coordinate observations
XV_OEOP	- Display of exterior orientation parameter observ.
XV_IMAGES	- Display of image data
XV_OPT	- Display of object point cordinates
XV_CAMERA	- Display of camera calibration data
XV_SOLEOP	- Display of eop results
XV_SOLOPC	- Display of opc results
XV_SOLRESID	- Display of residuals
XV_SOLNORMAL	- Display of normal equation system

6. Dependencies
- Is part of module FLOW_CON
- Interacts with the modules

  ADJUST
  REPORT

8. Resources None required.

3.7. FC_UPDATE - Interface to the ADJUST module

1. Identification FC_UPDATE

2. Type Module.

3. Purpose Provides an interface to the module ADJUST.

4. Function This module performs control of the operations in the ADJUST module (specially in the function AD_RELIN).

5. Subordinates This module is composed of the function:

FC_SEQ	- Update sequentially
FC_SIM	- Update and relinearize simultaneously

6. Dependencies
- Is part of module FLOW_CON
- Interacts with the module

  ADJUST

8. Resources None required.

3.8. FC_QUALCON - Interface to the QUAL_CON module

1. Identification FC_QUALCON

2. Type Module.

3. Purpose Provides an interface to the module QUAL_CON.

4. Function This module performs detailed control of the operations in the QUAL_CON module.

5. Subordinates This module contains the following functions:

BLUN_DET - Blunder detection
SELF_CALIB - Self calibration

6. Dependencies • Is part of module FLOW_CON

• Interacts with the modules

QUAL_CON

8. Resources None required.

3.9. FC_REPORT - Interface to the REPORT module

1. Identification FC_REPORT

2. Type Module.

3. Purpose Provides an interface to the module REPORT.

4. Function This module performs detailed control of the operations in the REPORT module.

5. Subordinates This module contains the following functions:

REP_DISK - Flow control for the saving data on disk
REP_PRINT - Flow control for the printing of reports.
REP_PLOT - Flow control for the plotting of data

6. Dependencies • Is part of module FLOW_CON

• Interacts with the module

REPORT

8. Resources None required.

4. INPUT - Data input

1. Identification INPUT

2. Type Module.

3. Purpose Receives input of data from outside the system and from measurements on digital images (Digital Photogrammetric Workstation), and organizes and stores it in internal data structures.

4. Function Functions of this module will control the input of data, and organize and store it in internal data structures. Data may come from the user dialog, from files, or.directly as measurements.

5. Subordinates This module is composed of the modules (See figure 2):

IN_IMAGE	- Input of digital images
IN_CON_PAR	- Input of control parameters
IN_OBJ_PAR	- Input of parameters for objects
IN_OBSERV	- Input of observation data

6. Dependencies (See Figure 1)

- The module is part of module OLTRIS
- The module interacts with the modules

FLOW_CON
USER_INT

DIG_IM
CONTR_PAR
MODEL_DATA

8. Resources Hardware : Disk
Software : Interface to external sources.

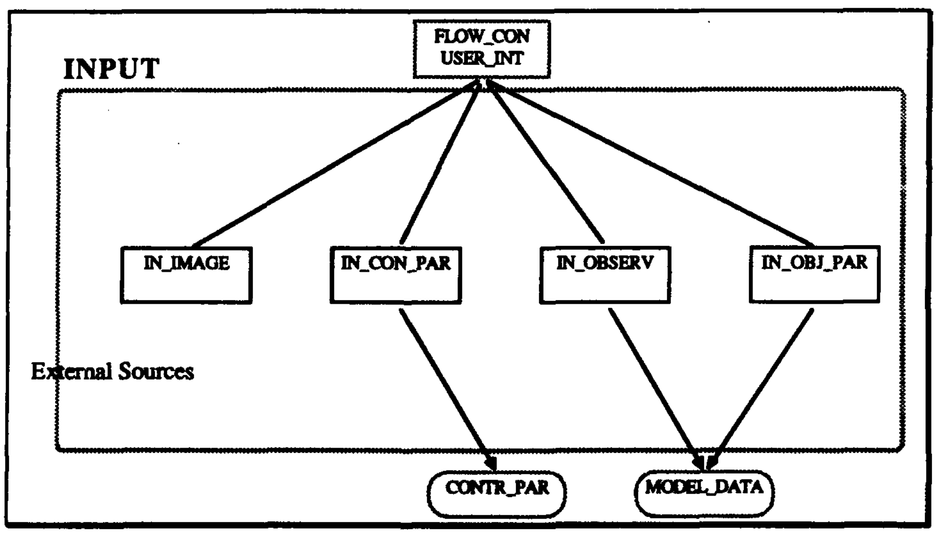

Figure 2: Overview of the INPUT module.

4.1. IN_IMAGE - Input of data from digital images

1. Identification IN_IMAGE

2. Type Module.

3. Purpose Handles the input of data from digital images.

4. Function This module contains functions necessary to read and store digital image data (image coordinate observations) in the system's internal storage.

(The first version of the input of digital images data into the system will use data sets consisting of lists of image coordinates measured on a digital workstation An extended version should provide an interface to an template matching program to organize the intermediate storage of this data in common data structures.)

5. Subordinates This module is composed of the modules (functions):

... (Not yet considered and installed)

6. Dependencies (See Figure 2)

- Is part of module INPUT
- Interacts with the modules

  FLOW_CON
  USER_INT/XVIEW

8. Resources Software: Interface to a template matching program.
Hardware: Digital photogrammetric workstation

4.2. IN_CON_PAR - Input of control parameters

1. Identification IN_CON_PAR

2. Type Module.

3. Purpose Handles the input of control parameters.

4. Function This module contains functions for fetching, organizing, and storing the control parameters. Data may come from manual input or from file.

5. Subordinates This module contains the following functions:

... (Not yet considered, because the input of control parameter comes directly from file or is set in Module USER_INT/XVIEW)

6. Dependencies (See Figure 2)

- Is part of module INPUT
- Interacts with the modules

  FLOW_CON
  USER_INT/XVIEW

  CONTR_PAR

8. Resources None required.

4.3. IN_OBJ_PAR - Input of parameters for objects

1. Identification IN_OBJ_PAR

2. Type Module.

3. Purpose Handle the input of parameters for objects.

4. Function This module contains functions for fetching, organizing, and storing parameters for various objects in the data structures of the system. Data may come from manual input or from file.

5. Subordinates This module is composed of the following.functions:

IN_OEOP - Input exterior orientation parameters
IN_OOPC - Input object point coordinates
IN_CAMERA - Input camera parameters
IN_IMG - Input image data

6. Dependencies (See Figure 2)

- Is part of module INPUT
- Interacts with the modules

FLOW_CON
USER_INT

MODEL_DATA

8. Resources Not yet considered.

4.4. IN_OBSERV - Input of observation data

1. Identification IN_OBSERV

2. Type Module.

3. Purpose Handle the input of observation data.

4. Function This module contains functions for fetching, organizing, and storing observation data in the data structures of the system. Data may come from manual input or from file.

5. Subordinates This module is composed of the functions:

IN_OIPC - Input image point coordinates observations
IN_OOPC - Input object point coordinates observations
IN_OEOP - Input exterior orientation parameters observations

6. Dependencies (See Figure 2)

- Is part of module INPUT
- Interacts with the modules

FLOW_CON
USER_INT

MODEL_DATA

8. Resources Not yet considered.

5. INITIAL - Initial parameter values computation

1. Identification INITIAL

2. Type Module.

3. Purpose Compute the initial values for all unknown parameters. Necessary because of the non-linear equation system.

4. Function Provide initial values of exterior orientation parameters and of object point coordinates for an aerial block using different methods of computing: intersection, resection in space and relative orientation.

5. Subordinates This module is composed of the modules:

INIT_EOP - Initial values for exterior orientation parameters
INIT_OPC - Initial values for object point coordinates

6. Dependencies • The module is part of module OLTRIS

• The module interacts with the modules

FLOW_CON
FUNC_LIB

CONTR_PAR
MODEL_DATA

8. Resources None required.

5.1. INIT_EOP - Initial values for exterior orientation

1. Identification INIT_EOP

2. Type Module.

3. Purpose Provide initial values for exterior orientation parameters.

4. Function This module contains functions for checking the presence of, and if necessary, computing (or otherwise provide) initial values for exterior orientation parameters.

5. Subordinates This module is composed of the two modules:

RELOR Functions for relative orientation
RESEC Functions for resection in space

6. Dependencies • Is part of module INITIAL

• Interacts with the modules

FLOW_CON
FUNC_LIB

CONTR_PAR
MODEL_DATA

8. Resources None required.

5.1.1. RELOR - Relative orientation

1. Identification RELOR

2. Type Module.

3. Purpose Compute the initial values of exterior orientation parameters by relative orientation by using the following method: coplanarity equations, single model.

4. Function Provide initial values of exterior orientation parameters and put it into the data structures

5. Subordinates This module is composed of the functions:

INIT_RELOR - checks some conditions for relative orientation
MOD_RELOR - relative orientation for models
LOCEOP_STARTVAL - provides start values to compute relor

6. Dependencies • The module is part of module INIT_EOP

• The module interacts with the modules

FLOW_CON
FUNC_LIB

CONTR_PAR
MODEL_DATA

8. Resources None required.

5.1.2. RESEC - Resection in space

1. Identification RESEC

2. Type Module.

3. Purpose Provide initial values for exterior orientation parameters.

4. Function This module contains functions for checking conditions of the resection, preparing data for the resection and executing the resection in space by an adjustment.

5. Subordinates This module is composed of the following functions:

INIT_EOP checks the conditions for resection in space
INIT_ABSEOP contains some functions for configuration tests
INIT_RESEC data preparation for resection in space
RES_APROX_VAL provides start values for resection in space
RES_COMP_RESEC adjustment of resection in space
RES_PAR_IPC computes the partial derivatives of observations

6. Dependencies • Is part of module INIT_EOP

• Interacts with the modules

FUNC_LIB

CONTR_PAR
MODEL_DATA

8. Resources None required.

5.2. INIT_OPC - Initial values for object point coordinates

1. Identification INIT_OPC

2. Type Module.

3. Purpose Provide initial values for object point coordinates.

4. Function This module contains functions for computing initial values for object point coordinates by an intersection. But the intersection requires two rays (images) to the point.

5. Subordinates This module is composed of the following functions:

INIT_OPC	- Checks conditions for an intersection
MAT_PROX_OPT	- Computation of approx. obj. point coordinates
MAT_INTERSEC	- Closed solution for intersection

6. Dependencies
- Is part of module INITIAL
- Interacts with the module

  FLOW_CON
  FUNC_LIB

  MODEL_DATA

8. Resources None required.

6. ADJUST - Bundle adjustment

1. Identification ADJUST

2. Type Module.

3. Purpose Perform the bundle adjustment tasks, simultaneous as well as sequential.

4. Function Build and factorize normal equations, update factorized normal equations sequentially, compute solution vector, and update the parameter vector.

5. Subordinates This module is composed of the following modules (See figure 3):

BUILD	- Build normal equations
CHFAC	- Factorize normal equations
UPDATE	- Update factorized normal equation system
SOLVE	- Solve normal equations
PARAM	- Update object parameters (unknowns)

6. Dependencies (See Figure 1)

- The module is part of module OLTRIS
- The module interacts with the modules

FLOW_CON
FUNC_LIB

CONTR_PAR
EQ_SYST
MODEL_DATA

8. Resources None required.

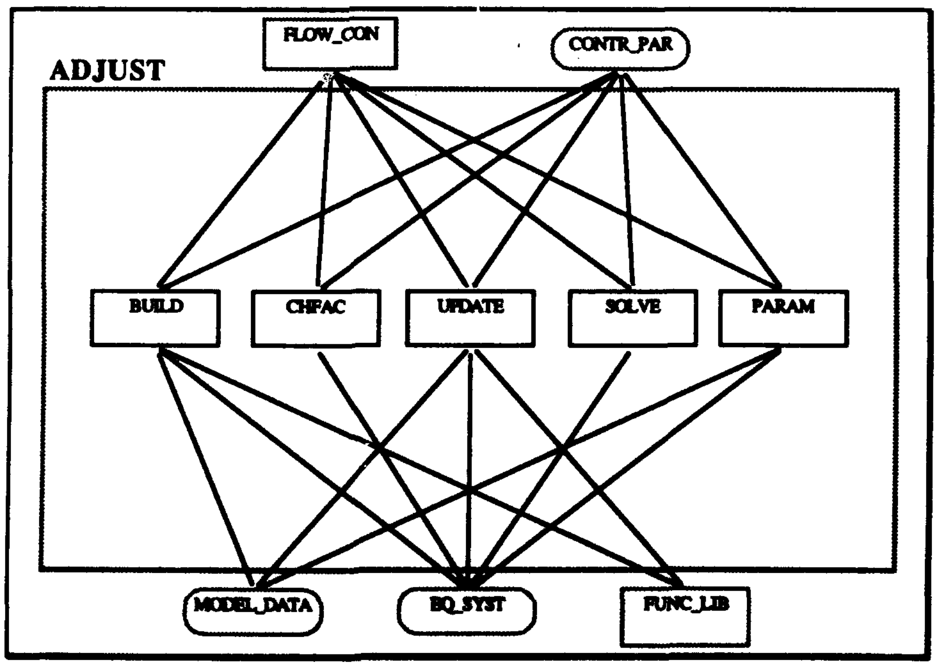

Figure 3: Overview of the ADJUST module.

6.1. BUILD - Build normal equations

1. Identification BUILD

2. Type Module.

3. Purpose Build the normal equations (A^TPA, A^TPl).

4. Function Go through all observation data and compute and add their contributions to the normal equation system.

5. Subordinates This module is composed of the following functions:

	Add contributions to the normals
SIM_BUILD	- One complete observation equation
SIM_BLD_IPC	- Observations of one image point
SIM_BLD_OPC	- Observations of obj. coord. of one obj. point
SIM_BLD_EOP	- Observations of ext. ori. param. of one image

6. Dependencies (See Figure 3)

- The module is part of module ADJUST
- The module interacts with the modules

FLOW_CON
FUNC_LIB

CONTR_PAR
EQ_SYST
MODEL_DATA

8. Resources None required.

6.2. CHFAC - Factorize normal equations

1. Identification CHFAC

2. Type Module.

3. Purpose Factorize the normal equation system, i.e. form an upper triangular matrix.

4. Function The normal equation system is factorized by Cholesky factorization.

5. Subordinates This module is composed of the following functions:

SIM_CHFAC	- Cholesky factorization (general)
SIM_CHFAC_HYP	- Chol. fact. (special hyper-row treatment)
SIM_CHFAC_RES	- Chol. fact. (resident part of the normals)

6. Dependencies (See Figure 3)

- The module is part of module ADJUST
- The module interacts with the modules

FLOW_CON

(CONTR_PAR)
EQ_SYST

8. Resources None required.

6.3. UPDATE - Update factorized normal equation system

1. Identification UPDATE

2. Type Module.

3. Purpose Update the factorized normal equation system with the contribution from one or more additional observations.

If "continuous re-linearization" is implemented, a module should be added, which will organize the "simultaneous" down- and update with old and new contributions from a specified set of observations.

4. Function Fetch the specified observation(s), compute the observation equation coefficients, and perform the update.

An algorithm based on Givens Transformation is used for the update.

5. Subordinates This module is composed of the functions:

UD_IPC	- Update by one image point observation
UD_1IPC	- Organize update with one image point observation
UD_SQIPC	- Organize updates with a sequence of image points
UD_OPIPC	- Organize updates with image points concerning one object point
UD_IMIPC	- Organize updates with sequence of image point observations from one image
UD_OPC	- Update by observations from one object point
UD_1OPC	- Organize update by observations from one object point
UD_EOP	- Update by observations of exterior orientation
UD_1EOP	- Organize update by observations of exterior orientation
GT_UPDATE	- Perform Givens Transformation update
GT_DODATE	- Perform Givens Transformation downdate

6. Dependencies (See Figure 3)

- Is part of module ADJUST
- Interacts with the module

FLOW_CON
FUNC_LIB

CONTR_PAR
EQ_SYST
MODEL_DATA

8. Resources None required.

6.4. SOLVE - Solve normal equations

1. Identification SOLVE

2. Type Module.

3. Purpose Solve the normal equation system, i.e. compute the corrections to the currently valid values of the parameters.

It should be possible to solve for a specified subset of the parameters, namely exterior orientation parameters for the last image(s), and object coordinates for specified points.

4. Function The normal equation system is solved by back substitution.

5. Subordinates This module is composed of the following modules:

SOL_BACKSUB - Back substitution
SOL_EOP - Back substitution for ext. ori. parameters
SOL_OPC - Back substitution for object point coordinates

6. Dependencies (See Figure 3)

- The module is part of module ADJUST
- The module interacts with the modules

FLOW_CON

(CONTR_PAR)
EQ_SYST

8. Resources None required.

6.5. PARAM - Update object parameters (unknowns)

1. Identification PARAM

2. Type Module.

3. Purpose Update the values of the unknown parameters.

4. Function The solution vector of the equation system is added to the current parameter vector. It should be checked that the solution vector is computed after the most recent update of the equation system.

Later versions of this module should be able to update only a specified subset of the parameters. Normally that will apply to parameters for recently included image(s) and object points, which are likely to have large corrections.

5. Subordinates This module is composed of the following functions:

AD_RELIN - Simultaneous adjustment
AD_PAR_UPDATE - Update of the parameter vector (unknowns)

6. Dependencies (See Figure 3)

- The module is part of module ADJUST
- The module interacts with the modules

FLOW_CON

(CONTR_PAR)
EQ_SYST
MODEL_DATA

8. Resources None required.

7. QUAL_CON - Quality control

1. Identification QUAL_CON

2. Type Module.

3. Purpose Evaluates the computed results, the observations, and the mathematical model used.

4. Function Computes some general statistics, characterizing the results as a whole, performs blunder detection, and checks determinability and significance of additional parameters.

5. Subordinates This module is composed of the modules:

GEN_STAT - General statistics
BLUNDER - Blunder detection
AP_SIGNIF - Significance of additional parameters
AP_DETERM - Determinability of additional parameters

6. Dependencies (See Figure 1)

- The module is part of module OLTRIS
- The module interacts with the modules

FLOW_CON

CONTR_PAR
EQ_SYST
MODEL_DATA

8. Resources Not yet considered.

7.1. GEN_STAT - General statistics

1. Identification GEN_STAT

2. Type Module

3. Purpose Computes general statistics from the adjustment.

4. Function Compute V^TPV, σ_0, redundancy, etc.

5. Subordinates This module contains the following function:

QC_GENSTAT - Computation of general statistics

6. Dependencies
- The module is part of module QUAL_CON
- The module interacts with the modules

FLOW_CON

EQ_SYST
ADJUST

8. Resources None required.

7.2. BLUNDER - Blunder detection

1. Identification BLUNDER

2. Type Module

3. Purpose Detects, localizes and initiates removal of possible blunders.

4. Function Test all observations for the presence of a gross error with the method of Baarda´s data-snooping. version is including . Later, an extended version should test a selection of observations to save computation time.

Along with research: Possible improvements or variants of data-snooping, as well as other methods will be implemented and tested.

5. Subordinates This module contains the following functions:

QC_SNOOP - Data snooping of all observations
QC_EXCLUDEBLUNDER - Exclude of one obs. from the normals

6. Dependencies • The module is part of module QUAL_CON

• The module interacts with the modules

FLOW_CON

CONTR_PAR
EQ_SYST
MODEL_DATA.

8. Resources None required.

7.3. AP_SIGNIF - Significance of additional parameters

1. Identification AP_SIGNIF

2. Type Module

3. Purpose Test the significance of the computed additional parameters, and initiate recalculation without the excluded parameters.

4. Function Not yet described.

5. Subordinates Not yet decomposed.Also this module may be composed of alternative modules, during research and test of methods.

6. Dependencies • The module is part of module QUAL_CON

• The module interacts with the module

FLOW_CON

CONTR_PAR
EQ_SYST
MODEL_DATA

8. Resources Not yet considered.

7.4. AP_DETERM - Determinability of additional parameters

1. Identification AP_DETERM

2. Type Module

3. Purpose Test the determinability of the additional parameters, and initiate recalculation without the excluded parameters.

4. Function Not yet described.

5. Subordinates Not yet decomposed.Also this module may be composed of alternative modules, during research and test of methods.

6. Dependencies • The module is part of module QUAL_CON

• The module interacts with the modules

FLOW_CON

CONTR_PAR
EQ_SYST
MODEL_DATA

8. Resources Not yet considered.

8. REPORT - Output of results

1. Identification REPORT

2. Type Module.

3. Purpose Format and output results; for displaying on the screen, as reports on paper, on secondary storage for further use. Plotting is included, on display (and on paper).

4. Function Separate modules will organize the handling of results. Display it on screen, print it on paper and store it in files on disk.

5. Subordinates This module is composed of the following modules:

RE_DISPLAY - Display results on the screen
RE_PRINT - Produce report on paper
RE_STORE - Store results on disk

6. Dependencies (See Figure 1)

• The module is part of module OLTRIS

• The module interacts with the modules

FLOW_CON

CONTR_PAR
MODEL_DATA

8. Resources Hardware:
- laser printer
- matrix printer
- disk
- screen of a workstation or terminal

8.1 RE_DISPLAY - Display of data and results on screen

1. Identification RE_DISPLAY

2. Type Module.

3. Purpose Display of input data (observations and objects), of intermediate results and the solution on the screen .

4. Function The displaying of the different data will be excuted in different functions .

5. Subordinates This module is composed of the following modules:

XV_DSP_OIPC	- Display of ipc obs. in measured sequence
XV_DSP_OIPC_IMG	- Display of ipc observations in image sequence
XV_DSP_OIPC_OPT	- Display of ipc obs. in object point sequence
XV_DSP_OOPC	- Display of object point coordinate observations
XV_DSP_OEOP	- Display of exterior orientation parameter obs.
XV_DSP_IMAGES	- Display of image data
XV_DSP_OPT	- Display of object point cordinates
XV_DSP_CAMERA	- Display of camera calibration data
XV_DSP_SOLEOP	- Display of eop results
XV_DSP_SOLOPC	- Display of opc results
XV_DSP_SOLRESID	- Display of residuals
XV_DSP_SOLNORMAL	- Display of normal equation system

6. Dependencies (See Figure 1)

- The module is part of module REPORT
- The module interacts with the modules

  FLOW_CON

  CONTR_PAR
  MODEL_DATA

8. Resources Hardware: screen of a workstation or terminal.

8.2 RE_PRINT - Output of results and data on printer

1. Identification RE_PRINT

2. Type Module.

3. Purpose Format and output results and data as reports and listings on paper. Plotting may be included in an extended version of the program.

Intermediate results for later resuming of the triangulation is capable to be stored in files on disk.

4. Function Functions provide an interface to the printer for output of results and data.

5. Subordinates This module is composed of the following functions:

XV_PR_FILE	- Print of a specific file
XV_PR_OIPC	- Print of ipc obs. in measured sequence
XV_PR_OOPC	- Print of object point coordinate observations
XV_PR_OEOP	- Print of exterior orientation parameter obs.
XV_PR_EOP	- Print of exterior orientation parameters
XV_PR_OPT	- Print of object point cordinates
XV_PR_CAMERA	- Print of camera calibration data
XV_PR_GENSTAT	- Print of general statistics
XV_PR_NORMAL	- Print of normal equation system

6. Dependencies (See Figure 1)

- The module is part of module REPORT
- The module interacts with the modules

  FLOW_CON

  CONTR_PAR
  MODEL_DATA

8. Resources Harware: laser printer
matrix printer
(plotter)

8.3 RE_STORE - Storage of data and results on disk

1. Identification RE_STORE

2. Type Module.

3. Purpose Storage of input data (observations and objects), of intermediate results and the solution on disk.

4. Function The storing of the different data will be excuted in different functions .

5. Subordinates This module is composed of the following functions:

ST_CAMERA - Saves camera calibration data in file
ST_PARAM - Saves control parameter in file

ST_RESULT - Saves the intermediate or final result in file
ST_EXTORI - Saves exterior orientation parameter in file
ST_OBJECT - Saves object point coordinates in file
ST_NORMAL - Saves the normal equation system in file

6. Dependencies (See Figure 1)

- The module is part of module REPORT
- The module interacts with the modules

FLOW_CON
EQ_SYST

CONTR_PAR
MODEL_DATA

8. Resources Hardware: disk for output.

9. FUNC_LIB - Library for common functions

1. Identification FUNC_LIB

2. Type Module.

3. Purpose Collection of functions used by different modules.

4. Function This module contains functions for computing mathematical applications, e.g. rotation matrix, design matrix coefficients, and for administrating common applications, e.g. common input of values, file-handling.

5. Subordinates This module is composed of the following modules:

FL_MATH - Module with mathematical functions
FL_UTIL - Module with utilities
FL_MATOP - Module with matrix operations
FL_TRAFO - Module with transformation functions
FL_TIME - Module with time functions

6. Dependencies (See Figure 1)

- The module is part of module OLTRIS
- The module interacts with the modules

INITIAL
ADJUST
FLOW_CON
REPORT

8. Resources None required.

9.1. FL_MATH - Mathematical functions

1. Identification FL_MATH

2. Type Module.

3. Purpose Collection of mathematical functions used by different modules.

4. Function This module contains a function for computing the elements of the rotation matrix, the determinant of the rotation matrix, an area of an oblique plane triangle and an area by the Gaussian area algorithm.

5. Subordinates This module is composed of the following functions:

M_ANGRMAT - Computes the elements of the rotation matrix
M_ANGRMAT_PHI - Comp. the elem. of the rot. mat. derived by phi
M_ANGRMAT_OME - Comp. the elem. of the rot. mat. deriv. by ome
M_ANGRMAT_KAP - Comp. the elem. of the rot. mat. deriv. by kap
M_ANGRMAT_TRANS - Computes transposed rotation matrix
M_ANGRMAT_TRANSPHI - Comp. transposed rot. mat. deriv. phi
M_ANGRMAT_TRANSOME - Comp. transposed rot. mat. deriv. ome
M_ANGRMAT_TRANSKAP - Comp. transposed rot. mat. deriv. kap
M_ANGLES - Computes angles from elements of rot. matrix
M_MULTMAT - Multiplicates two rotation matrices
M_DETRMAT - Computes the determinant of the rotation matrix
M_AREATRI - Computes an area of an oblique plane triangle
M_AREA - Computesan area by the Gaussian area algorithm

6. Dependencies • The module is part of module FUNC_LIB.

• Access from the following modules

ADJUST
INITIAL

8. Resources None required.

9.2. FL_UTIL - Utilities functions

1. Identification FL_UTIL

2. Type Module.

3. Purpose Collection of utilities functions used by different modules.

4. Function This module contains functions for input of integer, double values and words, also for opening of files for reading and writing, checking the presence of observations to avoid overwriting of data, finding observations in the buffer.

5. Subordinates This module is composed of the following functions:

INP_SKP - Cancel the rest of an line
INP_WRD - Input a single word
INP_INT - Input a single integer
INP_DBL - Input a single double

FILOPEN_R - Open a file for reading
FILOPEN_W - Open a file for writing
FILOPEN_CAMALL - Open all files of camera calibration data

INCHECK_OEOP - Checks the presence of eop-observations

FIND_FOIPC - find the oipc-observation in the buffer
FIND_FOOPC - find the oopc-observation in the buffer
FIND_FOEOP - find the oeop-observation in the buffer
FIND_IMG - find the image in the buffer

6. Dependencies
- The module is part of module FUNC_LIB.
- Access from following modules

  USER_INT
  INPUT
  FLOW_CON
  REPORT

8. Resources None required.

9.3. FL_MATOP - Matrices operation functions

1. Identification FL_MATOP

2. Type Module.

3. Purpose Collection of functions for matrices operation

4. Function This module contains functions for printing and handling of matrices used for 3D-Helmert transformation and also for relative orientation.

5. Subordinates This module is composed of the following functions:

ADJ_PRMS - Prints matrix on screen
ADJ_UPRMS - Prints upper triangular matrix on screen
ADJ_MATMUL - Multiplicats matrices
ADJ_UATAATL - Computes matrices sequentially
ADJ_UADJ2 - Solves normals by Gaussian factorization
ADJ_ULUDEC1 - LU decomposition of upper triangular matrix
ADJ_ULUSOL1 - Solves LU decomposed system
ADJ_UMATZERO - Sets all elements of upper triang. matrix to zero
ADJ_MATZERO -.Sets all elements of a full matrix to zero

ROTMAT - Computes rotation matrix (different way)
FORM_ABC - Forms partials
VEC_PROD - Multiplicates vectors
VEC_FORM - Changes storage of matrices
MAT_INV_PIV - Invertes a sym. fully stored matrix (pivoting)
FACT_BACK_INV - Factorizes, substitutes and invertes normals

6. Dependencies • The module is part of module FUNC_LIB.

• Access from following modules

INITIAL
FUNC_LIB

8. Resources None required.

9.4. FL_TRAFO - Library of functions for transformation

1. Identification FL_TRAFO

2. Type Module.

3. Purpose Collection of functions for 3D-Helmert transformation.

4. Function This module contains functions for preparing the data for the transformation, for computing the partial derivatives of each observation and for computing the seven parameter of transformation by an adjustment.

5. Subordinates This module is composed of the following functions:

T_ADPAR - Data preparation for transformation
T_TRAFO - 3D-Helmert transformation
T_PAR_OPC - Partials of each observation

6. Dependencies (See Figure 1)

• The module is part of module FUNC_LIB.

• Access from the modules

FLOW_CON

8. Resources None required.

9.5. FL_TIME - Time functions

1. Identification FL_TIME

2. Type Module.

3. Purpose Collection of functions for measuring the cpu, elapsed time.

4. Function This module contains functions to measure the cpu and/or elapsed time, to return the current date.

5. Subordinates This module is composed of the following functions:

TIMER_START_CPU	- Start the clock
TIMER_STOP_CPU	- Stop the clock and measure cpu-time
TIMER_START	- Start the clock
TIMER_STOP	- Stop the clock and measure the elapsed time
GET_CURRENT_DATE_INT	- Get the current date as integer
GET_CURRENT_DATE_CHR	- Get the current date as char

6. Dependencies • The module is part of module FUNC_LIB.

• Access from the following modules

ADJUST

8. Resources None required.

10. MEM_ALLOC - Memory allocation functions

1. Identification MEM_ALLOC

2. Type Module.

3. Purpose Collection of functions for memory allocation and reallocation for specific items of the data structures defined in the system.

4. Function This module contains functions which allocate and reallocate space for objects of specified types, defined in this system.

5. Subordinates This module is composed of the following modules:

MEM_OBS	- Memory allocation for observation structures
MEM_PAR	- Memory allocation for parameter structures
MEM_EQSYS	- Memory allocation for normal equation structures

6. Dependencies (See Figure 1)

• The module is part of module OLTRIS

• The module interacts with the modules

MODEL_DATA
EQ_SYST

8. Resources Hardware: - REM (Random excess memory).

10.1. MEM_OBS - Memory allocation for observation structures

1. Identification MEM_OBS

2. Type Module.

3. Purpose Collection of functions for memory allocation and release for the data structures of observations, defined in the system.

4. Function This module contains functions which allocate and release space for observations of specified types, defined in this system.

5. Subordinates This module is composed of the following functions:

MEM_OIPC - Memory allocation for image point observations
MEM_OOPC - Memory allocation for object point observations
MEM_OEOP - Memory allocation for ext. orient. param. observ.

MEMF_OIPC - Release of memory for image point observations
MEMF_OOPC - Release of memory for object point observations
MEMF_OEOP - Release of memory for ext. orient. param. observ.

6. Dependencies (See Figure 1)

- The module is part of module MEM_ALLOC
- The module interacts with the modules

  MODEL_DATA
  INPUT

8. Resources Hardware: - REM (Random excess memory).

10.2. MEM_PAR - Memory allocation for parameter structures

1. Identification MEM_PAR

2. Type Module.

3. Purpose Collection of functions for memory allocation and release of memory for data structures of parameters defined in the system.

4. Function This module contains functions which allocate and release space for parameters of specified types, defined in this system.

5. Subordinates This module is composed of the following functions:

MEM_OPT - Memory allocation for object point item
MEM_OPC - Memory allocation for object point coordinates
MEM_EOP - Memory allocation for ext. orient.
MEM_IMG - Memory allocation for observation structures
MEM_CAMERA - Memory allocation for parameter structures

MEMF_OPT - Release of memory for object point item
MEMF_OPC - Release of memory for object point coordinates
MEMF_EOP - Release of memory for ext. orient.
MEMF_IMG - Release of memory for observation structures
MEMF_CAMERA - Release of memory for parameter structures

6. Dependencies (See Figure 1)

- The module is part of module MEM_ALLOC
- The module interacts with the modules

MODEL_DATA
INPUT

8. Resources Hardware: - REM (Random excess memory).

10.3. MEM_EQSYS - Mem. alloc. for normal equation structures

1. Identification MEM_EQSYS

2. Type Module.

3. Purpose Collection of functions for memory allocation, reallocation and release of memory for specific items of the normal equation data structures defined in the system.

4. Function This module contains functions which allocate, reallocate and release space for objects of normal equation system, defined in this system.

5. Subordinates This module is composed of the following functions:

MEM_UPPER - Memory allocation for upper triangle of matrix
MEM_VECTOR - Memory allocation for column or row vector
MEM_AIPC - Memory allocation for design submatrix of ipc
MEM_AOPC - Memory allocation for design submatrix of opc
MEM_AEOP - Memory allocation for design submatrix of eop
MEM_UXT - Memory allocation for submatrix in the hyperrow
MEM_HYP - Memory allocation for hyperline in the normals

MEMR_UPPER - Memory reallocation for upper triangle of matrix
MEMR_VECTOR - Memory reallocation for column or row vector

MEMF_UPPER - Release of memory for upper triangle of matrix
MEMF_VECTOR - Release of memory for column or row vector
MEMF_AIPC - Release of memory for design submatrix of ipc
MEMF_AOPC - Release of memory for design submatrix of opc
MEMF_AEOP - Release of memory for design submatrix of eop
MEMF_UXT - Release of memory for submatrix in the hyperrow
MEMF_HYP - Release of memory for hyperline in the normals

6. Dependencies (See Figure 1)

- The module is part of module MEM_ALLOC
- The module interacts with the modules

MODEL_DATA
EQ_SYST

8. Resources Hardware: - REM (Random excess memory).

11. CONTR_PAR - Control parameters

1. Identification CONTR_PAR

2. Type Data module.

3. Purpose To contain control parameters, i.e. all parameters and data used for controlling the different functions performed by the system.

4. Function The module contains the initialization of control flow parameters, as well as critical values and other general parameters needed by some computations.

5. Subordinates This module contains the following functions:

CP_ADJUST - Control parameters for the adjustment
CP_EQSYS - Control parameters for the normal equation system
CP_MISC - Control parameters for the miscellaneous applications

6. Dependencies • This module is part of module OLTRIS

• Interacts probably with all high-level functional modules.

8. Resources The data is supposed to reside in core.

12. MODEL_DATA - Data for entities of the mathematical model

1. Identification MODEL_DATA

2. Type Data module.

3. Purpose Organize storage of data concerning entities used in the mathematical model.

4. Function This module contains data structures for the "physical" entities cameras, images, and object points; for a priori and computed parameters, and for observations and design matrix.

5. Subordinates This module is composed of the modules (see figure 4):

OPT - Object point data
OPC - Object point coordinates
EOP - Exterior orientation parameters
IMAGE - Image data
CAMERA - Camera data
OIPC - Observations of image point coordinates
OOPC - Observations of object point coordinates
OEOP - Observations of exterior orientation parameters

MD_FUNC - Functions for handling the data structures

6. Dependencies • This module is part of module OLTRIS

• Supposed to be accessed from the modules

INPUT
INITIAL
ADJUST
QUAL_CON
REPORT

• Uses

MEM_ALLOC

• By pointers and indices cross-related to data in

EQ_SYST

8. Resources The data is supposed to reside in core.

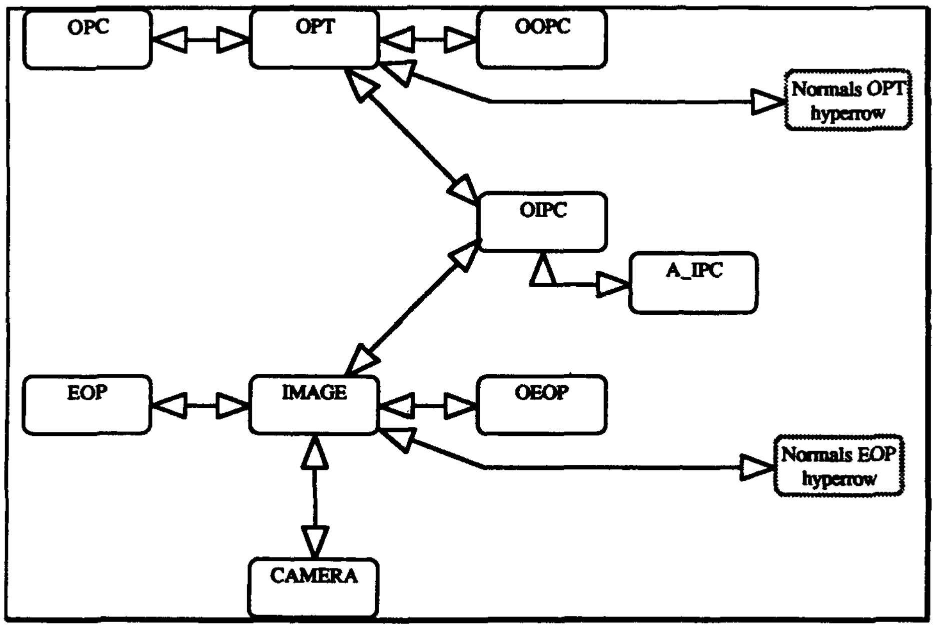

Figure 4: Relations between the data modules of MODEL_DATA. Two or one arrowheads indicate whether each instance of the entity at the other end of the line may be related to several or only one instance of the entity pointed at. The greyed boxes belong to module EQ_SYST.

12.1. MD_FUNC - Functions for handling the data structures

1. Identification MD_FUNC

2. Type Module.

3. Purpose Organize the handling of the model_data structures.

4. Function This module contains functions for input of data into the model_data structures and for finding data in its lists.

5. Subordinates This module is composed of the following functions:

MDI_OIPC	- Input observations of image point coordinates into list
MDI_OOPC	- Input observations of object point coordinates into list
MDI_OEOP	- Input observations of ext. orientation param. into list
MDI_OPT	- Input object point coordinates into list
MDI_IMG	- Input image data into list
MDI_CAMERA	- Input camera data into list
MDF_OIPC	- Find observations of image point coordinates in list
MDF_OOPC	- Find observations of object point coordinates in list
MDF_OEOP	- Find observations of ext. orientation parameters in list
MDF_OPT	- Find object point coordinates in list
MDF_IMG	- Find image data in list
MDF_CAMERA	- Find camera data in list

6. Dependencies
- This module is part of module MODEL_DATA.
- Access from the modules

  INPUT
  INITIAL
  ADJUST
  QUAL_CON
  REPORT
- Uses
  data modules of MODEL_DATA

8. Resources None required.

13. EQ_SYST - Normal equation system and related data

1. Identification EQ_SYST

2. Type Data module.

3. Purpose Organize storage for the normal equation system.

4. Function This module contains data comprising the normal equation system, the elements of which are coefficient matrix, constant vector(s) and solution vector. Also included is related work-space, e.g. for observation equations during sequential updates.

5. Subordinates This module is composed of the modules:

EQ_BASE - Basic parts of the normal equations system
EQ_HYPER - Hyper-row parts of the normal equations system
EQ_WORKSP - Work-space related to the normal equations system

EQ_FUNC - Functions for handling the normal equations system

6. Dependencies • This module is part of module OLTRIS

• Access from the modules

ADJUST
QUAL_CON

• Uses

MEM_ALLOC

8. Resources To achieve minimal response time, this data should reside permanently in core. If possible it should be prevented from any swapping / paging.

13.1. EQ_BASE - Basic parts of the normal equations system

1. Identification EQ_BASE

2. Type Data module.

3. Purpose Organize storage for the basic parts of the normal equation system.

4. Function This module contains data comprising the basic part of the normal equation system, the elements of which are coefficient matrix, constant vector(s) and solution vector.

5. Subordinates This module is composed of the following data structures:

A_IPC - Design submatrix for image point observation
A_OPC - Design submatrix for object point observation
A_EOP - Design submatrix for ext. orient. param. observation

6. Dependencies • This module is part of module EQ_SYST.

• Access from the modules

ADJUST
QUAL_CON

• Uses

MEM_ALLOC

8. Resources To achieve minimal response time, this data should reside permanently in core. If possible it should be prevented from any swapping / paging.

13.2. EQ_HYPER - Hyperrow parts of the normal equat. system

1. Identification EQ_HYPER

2. Type Data module.

3. Purpose Organize storage for the hyperrow parts of the normal equation system.

4. Function This module contains data comprising the hyperrow parts of the normal equation system, the elements of which are coefficient matrix, constant vector(s) and solution vector.

5. Subordinates This module is composed of the data structures:

HYP_OPC - Hyperline in the normals
SUB_XT - Submatrix in the object point hyperrow

6. Dependencies
- This module is part of module EQ_SYST.
- Access from the modules

  ADJUST
  QUAL_CON
- Uses

  MEM_ALLOC

8. Resources To achieve minimal response time, this data should reside permanently in core. If possible it should be prevented from any swapping / paging.

13.3. EQ_WORKSP - Workspace related to the normal eq. syst.

1. Identification EQ_WORKSP

2. Type Data module.

3. Purpose Organize storage of related data/variables for the normal equation system.

4. Function This module contains data comprising related data/variables of the normal equation system. Also included is allocated work-space of this gobal variables.

5. Subordinates This module is composed of the following data (variables):

IMAGE_INIT - Initial number of images for allocation
IMAGE_ADD - Number of additional images at reallocation
EQ_PARAMS - Number of image parameter per image
N_AWSP - Number of lines obs. eq. workspace to allocate
SIZE_HYP - Number of lines to allocate in a hyperrow
EQ_IND - Compute 'linear' index in an 'Upper' matrix

6. Dependencies
- This module is part of module EQ_SYST.
- Access from the modules

  ADJUST
  QUAL_CON

8. Resources To achieve minimal response time, this data should reside permanently in core. If possible it should be prevented from any swapping / paging.

13.4. EQ_FUNC - Functions for handling the normal eq. system

1. Identification EQ_FUNC

2. Type Module.

3. Purpose Organize administrating of the normal equation system.

4. Function This module contains functions for administrating and house-keeping of the normal equation system.

5. Subordinates This module is composed of the following functions:

EQ_INIT	- Initialize the normal equation system storage
EQ_EXPAND	- Expand the normals
EQ_CLEAR	- Clear the normal equations system
EQ_SETHYP	- Basic parts of the normal equations system
EQ_ZERO	- Zero-fill the current normal equations system
EQ_PRINT	- Print the current normal equations system
EQ_SIZEAL	- Return sizes of currently allocated space for normals
EQ_SIZCU	- Return sizes of the current normal equations system
EQ_POINTR	- Return pointers to normal equation system storage
EQ_HYPLOAD	- Load a hyperrow from separate storage
EQ_HYPSAVE	- Save a hyperrow to separate storage

6. Dependencies
- This module is part of module EQ_SYST.
- Interacts with the following modules

  ADJUST
  QUAL_CON

8. Resources None required.